argo data management

Argo data management

March 1, 2006

ar-um-02-01

# USER'S MANUAL

## Version 2.1

Argo data management

User's manual

Ref : ar-um-02-01
Ref ifremer : cor-do/dti-mut/02-084

Version : 2.1

Date : 01/03/2006

Authors : Thierry Carval / Ifremer, Bob Keeley / MEDS, Yasushi Takatsuki / JAMSTEC, Takashi Yoshida / JMA, Stephen Loch / BODC, Claudia Schmid / AOML, Roger Goldsmith / WHOI, Annie Wong / UW, Rebecca McCreadie / BODC, Ann Thresher / CSIRO, Ahn Tran / MEDS

# Table of contents

# History of the document

| Version | Date | Comment |
|---|---|---|
| 0.9 | 29/12/2001 | Thierry Carval : creation of the document |
| 0.9a | 18/01/2001 | Bob Keeley : general comments and updates |
| 0.9a | 24/01/2002 | Valérie Harscoat : general comments and updates |
| 0.9a | 25/01/2002 | Claudia Schmid : general comments and updates |
| 0.9a | 24/01/2002 | Roger Goldsmith : general comments and updates |
| 0.9b | 05/03/2002 | Roger Goldsmith, Yasushi Takatsuki and Claudia Schmid comments implemented. |
| 0.9c | 24/04/2002 | Comments from version 0.9b are implemented |
| 1.0 | 09/07/2002 | Comments from version 0.9c are implemented |
| 1.0a | 31/12/2002 | Missing values in trajectory and calibration |
| 1.0a | 17/01/2003 | Description of directory file format |
| 1.0a | 24/01/2003 | Update of reference tables |
| 1.0a | 24/01/2003 | Update of "measurements of each profile" to handle corrected values |
| 1.0a | 24/01/2003 | Increase the size of DC_REFERENCE from STRING16 to STRING32 |
| 1.0b | 17/03/2003 | Replace corrected values with adjusted values |
| 1.0b | 29/04/2003 | DC_REFERENCE removed from trajectory format general information of the float section |
| 1.0b | 30/04/2003 | Use blank fill values for character variables |
| 1.0c | 30/04/2003 | Proposal submitted on 30/04/2003 |
| 1.0d | 14/08/2003 | Proposal submitted on 14/08/2003 (green font) |
| 1.0e | 23/10/2003 | Proposal submitted on 12/11/2003 (green font) |
| 2.0 | 12/11/2003 | All comments from "Argo user's manual comments" ref ar-dm-02-02 implemented.<br>General agreement from Argo data management meeting in Monterey (Nov. 5-7, 2003) |
| 2.01 | 15/12/2003 | History section updated. |
| 2.01 | 01/10/2004 | Meta-data section :<br>WMO_INST_TYPE added to history section<br>INSTRUMENT_TYPE renamed INST_REFERENCE |
| 2.01 | 10/11/2004 | Reference table 2 quality control flag scale updated by Annie Wong |
| 2.01 | 10/11/2004 | Updates in reference table 3, parameter codes table<br>DOXY, TEMP_DOXY, TEMP (use ITS-90 scale) |
| 2.01 | 23/11/2004 | Reference table 14 : instrument failure mode added by Annie Wong |
| 2.01 | 25/02/2005 | Table 11 updated for frozen profile and deepest pressure tests from Rebecca Macreadie |
| 2.01 | 28/02/2005 | Table 4 updated : CSIO, China Second Institute of Oceanography |

| | | |
|---|---|---|
| 2.01 | 12/04/2005 | Mathieu Belbeoch : table 5 updated : argos location classes |
| 2.01 | 12/06/2005 | Change lengths of all parameter name variables to accomodate longer parameter names. Affects: STATION_PARAMETERS (section 2.2.3), PARAMETER (section 2.2.5), and HISTORY_PARAMETER (section 2.2.6) in the profile format; TRAJECTORY_PARAMETERS (section 2.3.3) and HISTORY_PARAMETER (section 2.3.6) in the trajectory format; SENSOR (section 2.4.5) and PARAMETER (section 2.4.6) in the meta-data format |
| 2.01 | 12/06/2005 | Change ":conventions" attribute and description of PROFILE_<PARAM>_QC in section 2.2.3. |
| 2.01 | 12/06/2005 | Add reference table 2a for the redefined PROFILE_<PARAM>_QC variables |
| 2.01 | 20/06/2005 | New long name for TEMP_DOXY in section 3.3 |
| 2.01 | 22/06/2005 | Claudia Schmid : general update of trajectory file history section (N_MEASUREMENT dimension removed) |
| 2.01 | 07/11/2005 | Claudia Schmid : create reference table 14 for technical parameter names.<br>Minor typo corrections. |
| 2.01 | 07/11/2005 | Thierry Carval : add a GPS code for position accuracy in ref. Table 5. |
| 2.01 | 08/11/2005 | Ann Thresher : exemple of sensor type in meta-data |
| 2.01 | 09/11/2005 | Annie Wong : §3.2.2 usage of <PARAM_ADJUSTED_QC> and <PARAM_QC><br>Reference table 2 updated (qc 3 and 4) |
| 2.01 | 11/11/2005 | Thierry Carval : §2.2.4, §2.3.4 accept adjusted parameters in real time files |
| 2.01 | 11/11/2005 | Thierry Carval : §2.2.6 history section for multi-profile files is empty |
| 2.01 | 11/11/2005 | Thierry Carval : §1.3, §2.2.3, §2.3.4 real-time adjusted data |
| 2.01 | 11/11/2005 | Thierry Carval : §2.4.8 highly desirable meta-data description |
| 2.1 | 30/11/2005 | Annie Wong : §3.2.1 update on flag 4 real time comment |
| 2.1 | 20/12/2005 | Thierry Carval : remove erroneous blanks (ex : " Argo reference table 3" |
| 2.1 | 01/03/2006 | Mark Ignaszewski: §2.3.6 Change HISTORY_*_INDEX to "int", Change HISTORY_REFERENCE to STRING64. Change to "dependent" in all sections. Remove PLATFORM_SERIAL_NO from desirable parameter table. Add "No QC performed" to Table 2a. Change FORMAT_VERSION to 2.2 in all sections. |

# 1. Introduction

This document is the Argo data user's manual.

It contains the description of the formats and files produced by the Argo DACs.

## 1.1. Argo program, data management context

The objective of Argo program is to operate and manage a set of 3000 floats distributed in all oceans, with the vision that the network will be a permanent and operational system.

The Argo data management group is creating a unique data format for internet distribution to users and for data exchange between national data centres (DACs) and global data centres (GDACs).

Profile data, metadata, trajectories and technical data are included in this standardization effort.

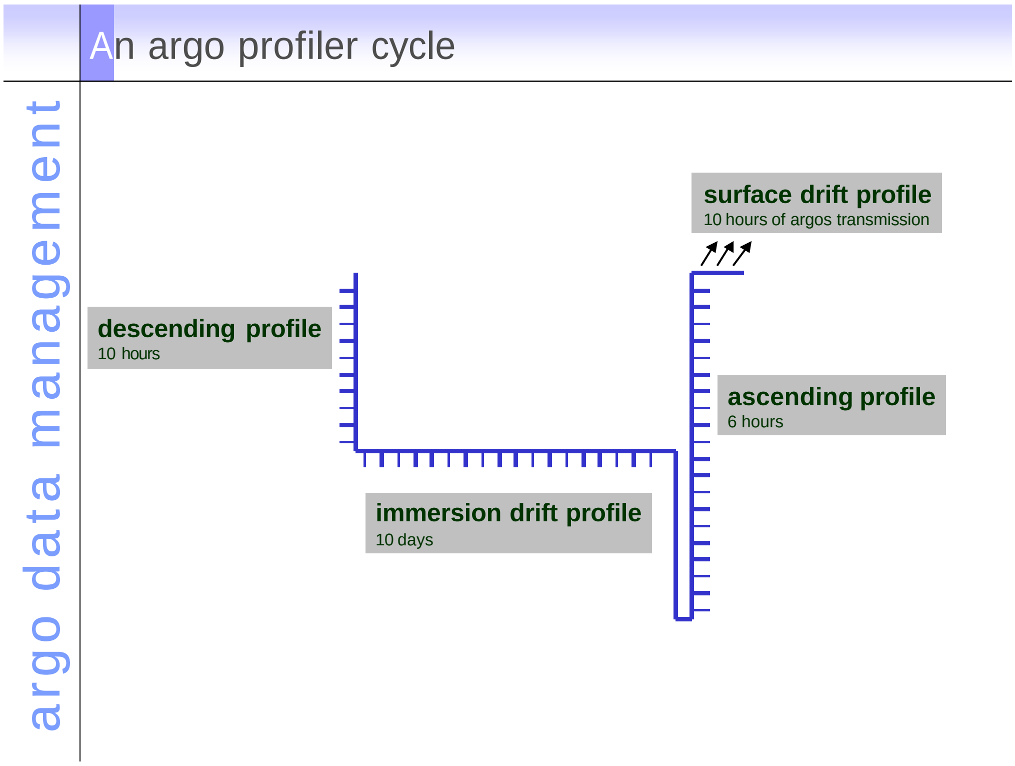
The Argo data formats are based on NetCDF because :

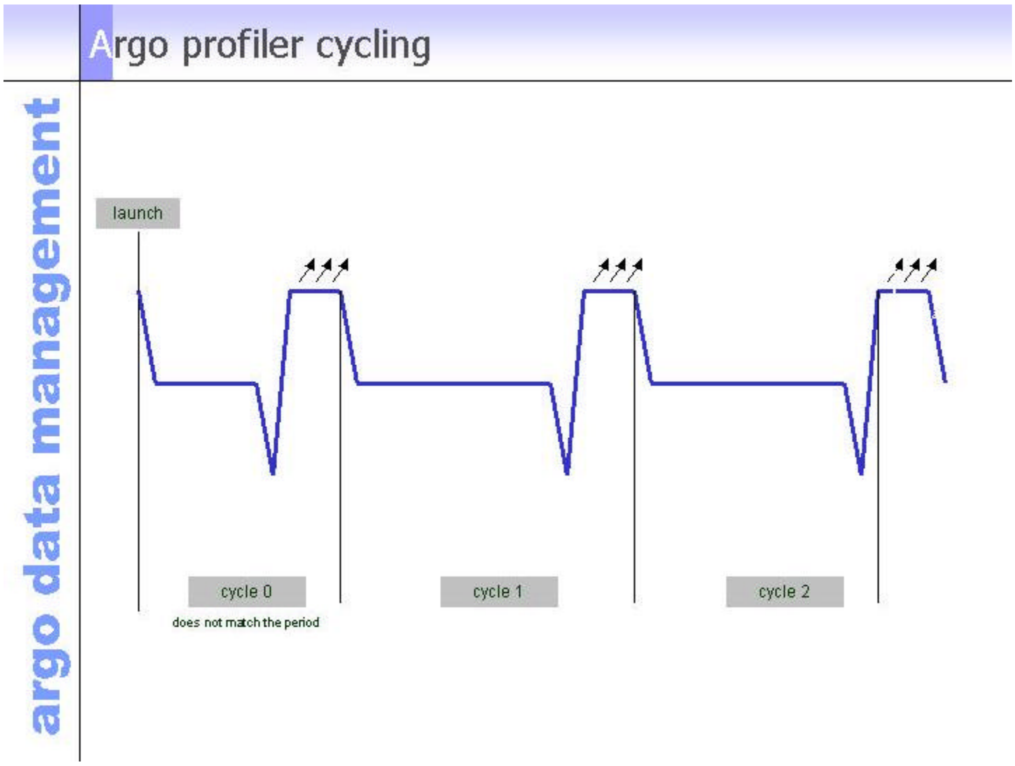
- It is a widely accepted data format by the user community,
- It is a self-describing format for which tools are widely available,
- It is a reliable and efficient format for data exchange.

## 1.2. Argo float data

An Argo float drifts for a number of years in the ocean. It continuously performs measurement cycles. Each cycle lasts about 10 days and can be divided into 4 phases :

- A descent from surface to a defined pressure (eg : 1500 decibars),
- A subsurface drift (eg : 10 days),
- An ascending profile with measurements (eg : pressure, temperature, salinity),
- A surface drift with data transmission to a communication satellite.




Some Argo floats start sending a cycle 0 with a shorter cycle time than the next cycles. Some other floats directly start their mission with a cycle 1 (standard cycle).

## 1.3. Real-time and Delayed mode data

Data from Argo floats are transmitted from the float, passed through processing and automatic quality control procedures as quickly as possible after the float begins reporting at the surface. The target is to issue the data to the GTS and Global Data servers within 24 hours of surfacing, or as quickly thereafter as possible. These are called real-time data.

The data are also issued to the Principle Investigators on the same schedule as they are sent to the Global servers. These scientists apply other procedures to check data quality and the target is for these data to be returned to the global data centres within 6 to 12 months. These constitute the delayed mode data.

The adjustments applied to delayed-data may also be applied to real-time data, to correct sensor drifts for real-time users. However, these real-time adjustments will be recalculated by the delayed mode quality control.

# 2. Formats description

## 2.1. Overview of the formats

Argo data formats are based on NetCDF from Ucar.

NetCDF (network Common Data Form) is an interface for array-oriented data access and a library that provides an implementation of the interface. The NetCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data. The NetCDF software was developed at the Unidata Program Centre in Boulder, Colorado. The freely available source can be obtained as a compressed tar file or a zip file from Unidata or from other mirror sites.

- Ucar web site address : http://www.ucar.edu/ucar
- NetCDF documentation : http://www.unidata.ucar.edu/packages/netcdf/index.html

Argo formats are divided in 4 sections :

- Dimensions and definitions
- General information
- Data section
- History section

The Argo NetCDF formats do not contain any global attribute.

Argo date and times : all date and time have to be given in UTC time, universal time coordinates.

## 2.2. Argo profile file format

An Argo profile file contains a set of profiles. The minimum number is one profile. There is no defined maximum number of profiles.

A profile contains measurements performed at different pressures by an Argo float.

A profile contains typically 100 pressures, from 0 decibar (surface) to 2000 decibars (approximately 2000 meters depth).

For each pressure sample, there is a fixed number of parameters measured or calculated such as temperature, salinity or conductivity.

For file naming conventions, see §4.1 .

### 2.2.1. Dimensions and definitions

| Name | Value | Definition |
|---|---|---|
| DATE_TIME | DATE_TIME = 14; | This dimension is the length of an ASCII date and time value.<br>Date_time convention is : YYYYMMDDHHMISS<br>• YYYY : year<br>• MM : month<br>• DD : day<br>• HH : hour of the day (as 0 to 23)<br>• MI : minutes (as 0 to 59)<br>• SS : seconds (as 0 to 59)<br>Date and time values are always in universal time coordinates (UTC).<br>Examples :<br>20010105172834 : January 5th 2001 17:28:34<br>19971217000000 : December 17th 1997 00:00:00 |
| STRING256<br>STRING64<br>STRING32<br>STRING16<br>STRING8<br>STRING4<br>STRING2 | STRING256 = 256;<br>STRING64 = 64;<br>STRING32 = 32;<br>STRING16 = 16;<br>STRING8 = 8;<br>STRING4 = 4;<br>STRING2 = 2; | String dimensions from 2 to 256. |
| N_PROF | N_PROF = <int value>; | Number of profiles contained in the file.<br>This dimension depends on the data set.<br>A file contains at least one profile.<br>There is no defined limit on the maximum number of profiles in a file.<br>Example :<br>N_PROF = 100 |
| N_PARAM | N_PARAM = <int value> ; | Maximum number of parameters measured or calculated for a pressure sample.<br>This dimension depends on the data set.<br>Examples :<br>(pressure, temperature) : N_PARAM = 2<br>(pressure, temperature, salinity) : N_PARAM = 3<br>(pressure, temperature, conductivity, salinity) : N_PARAM = 4 |
| N_LEVELS | N_LEVELS = <int value> ; | Maximum number of pressure levels contained in a profile.<br>This dimension depends on the data set.<br>Example : N_LEVELS = 100 |
| N_CALIB | N_CALIB = <int value> ; | Maximum number of calibrations performed on a profile.<br>This dimension depends on the data set.<br>Example : N_CALIB = 10 |
| N_HISTORY | N_HISTORY = UNLIMITED; | Number of history records. |

### 2.2.2. General information on the profile file

This section contains information about the whole file.

| Name | Definition | Comment |
|---|---|---|
| DATA_TYPE | char DATA_TYPE(STRING16);<br>DATA_TYPE:comment = "Data type";<br>DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file.<br>The list of acceptable data types is in the reference table 1.<br>Example : Argo profile |
| FORMAT_VERSION | char FORMAT_VERSION(STRING4);<br>FORMAT_VERSION:comment = "File format version";<br>FORMAT_VERSION:_FillValue = " "; | File format version<br>Example : «2.2» |
| HANDBOOK_VERSION | char HANDBOOK_VERSION(STRING4);<br>HANDBOOK_VERSION:comment = "Data handbook version";<br>HANDBOOK_VERSION:_FillValue = " "; | Version number of the data handbook.<br>This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook.<br>Example : «1.0» |
| REFERENCE_DATE_TIME | char REFERENCE_DATE_TIME(DATE_TIME);<br>REFERENCE_DATE_TIME:comment = "Date of reference for Julian days";<br>REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMISS";<br>REFERENCE_DATE_TIME:_FillValue = " "; | Date of reference for julian days.<br>The recommended reference date time is<br>"19500101000000" : January 1st 1950 00:00:00 |

### 2.2.3. General information for each profile

This section contains general information on each profile.

Each item of this section has a N_PROF (number of profiles) dimension.

| Name | Definition | Comment |
|---|---|---|
| PLATFORM_NUMBER | char PLATFORM_NUMBER(N_PROF, STRING8);<br>PLATFORM_NUMBER:long_name = "Float unique identifier";<br>PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII";<br>PLATFORM_NUMBER:_FillValue = " "; | WMO float identifier.<br>WMO is the World Meteorological Organization.<br>This platform number is unique.<br>Example : 6900045 |
| PROJECT_NAME | char PROJECT_NAME(N_PROF, STRING64);<br>PROJECT_NAME:comment = "Name of the project";<br>PROJECT_NAME:_FillValue = " "; | Name of the project which operates the profiling float that performed the profile.<br>Example : GYROSCOPE (EU project for ARGO program) |
| PI_NAME | char PI_NAME (N_PROF, STRING64);<br>PI_NAME:comment = "Name of the principal investigator";<br>PI_NAME:_FillValue = " "; | Name of the principal investigator in charge of the profiling float.<br>Example : Yves Desaubies |
| STATION_PARAMETERS | char STATION_PARAMETERS(N_PROF, N_PARAM,STRING16);<br>STATION_PARAMETERS:long_name = "List of available parameters for the station";<br>STATION_PARAMETERS:conventions = "Argo reference table 3";<br>STATION_PARAMETERS:_FillValue = " "; | List of parameters contained in this profile.<br>The parameter names are listed in reference table 3.<br>Examples : TEMP, PSAL, CNDC<br>TEMP : temperature<br>PSAL : practical salinity<br>CNDC : conductvity |
| CYCLE_NUMBER | int CYCLE_NUMBER(N_PROF);<br>CYCLE_NUMBER:long_name = "Float cycle number";<br>CYCLE_NUMBER:conventions = "0..N, 0 : launch cycle (if exists), 1 : first complete cycle";<br>CYCLE_NUMBER:_FillValue = 99999; | Float cycle number.<br>A profiling float performs cycles. In each cycle, it performs an ascending vertical profile, a subsurface drift and a surface drift. In some cases, it also performs a descending vertical profile.<br>0 is the number of the launch cycle. The |

| | | |
|---|---|---|
| | | subsurface drift of the cycle 0 may not be complete.<br>1 is the number of the first complete cycle.<br>Example :<br>10 : cycle number 10 |
| DIRECTION | char DIRECTION(N_PROF);<br>DIRECTION:long_name = "Direction of the station profiles";<br>DIRECTION:conventions = "A: ascending profiles, D: descending profiles ";<br>DIRECTION:_FillValue = " "; | Type of profile on which measurement occurs.<br>A : ascending profile<br>D : descending profile |
| DATA_CENTRE | char DATA_CENTRE(N_PROF, STRING2);<br>DATA_CENTRE:long_name = "Data centre in charge of float data processing";<br>DATA_CENTRE:conventions = "Argo reference table 4";<br>DATA_CENTRE:_FillValue = " "; | Code for the data centre in charge of the float data management.<br>The data centre codes are described in the reference table 4.<br>Example : ME for MEDS |
| DATE_CREATION | char DATE_CREATION(DATE_TIME);<br>DATE_CREATION:comment = "Date of file creation ";<br>DATE_CREATION:conventions = "YYYYMMDDHHMISS";<br>DATE_CREATION:_FillValue = " "; | Date and time (UTC) of creation of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011229161700 : December 29th 2001 16 :17 :00 |
| DATE_UPDATE | char DATE_UPDATE(DATE_TIME);<br>DATE_UPDATE:long_name = "Date of update of this file";<br>DATE_UPDATE:conventions = "YYYYMMDDHHMISS";<br>DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 09 :05 :00 |
| DC_REFERENCE | char DC_REFERENCE(N_PROF, STRING32);<br>DC_REFERENCE:long_name = "Station unique identifier in data centre";<br>DC_REFERENCE:conventions = "Data centre convention";<br>DC_REFERENCE:_FillValue = " "; | Unique identifier of the profile in the data centre.<br>Data centres may have different identifier schemes.<br>DC_REFERENCE is therefore not unique across data centres. |
| DATA_STATE_INDICATOR | char DATA_STATE_INDICATOR(N_PROF, STRING4);<br>DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through";<br>DATA_STATE_INDICATOR:conventions = "Argo reference table 6";<br>DATA_STATE_INDICATOR:_FillValue = " "; | Degree of processing the data has passed through.<br>The data state indicator is described in the reference table 6. |
| DATA_MODE | char DATA_MODE(N_PROF);<br>DATA_MODE:long_name = "Delayed mode or real time data";<br>DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment";<br>DATA_MODE:_FillValue = " "; | Indicates if the profile contains real time, delayed mode or adjusted data.<br>R : real time data<br>D : delayed mode data<br>A : real time data with adjusted values |
| INST_REFERENCE | char INST_REFERENCE(N_PROF, STRING64);<br>INST_REFERENCE:long_name = "Instrument type";<br>INST_REFERENCE:conventions = "Brand, type, serial number";<br>INST_REFERENCE:_FillValue = " "; | References of the instrument : brand, type, serial number<br>Example : APEX-SBE 259 |
| WMO_INST_TYPE | char WMO_INST_TYPE(N_PROF, STRING4);<br>WMO_INST_TYPE:long_name = "Coded instrument type";<br>WMO_INST_TYPE:conventions = "Argo reference table 8"; | Instrument type from WMO code table 1770.<br>A subset of WMO table 1770 is documented in the reference table 8.<br>Example :<br>846 : Webb Research float, Seabird sensor |

| | | |
|---|---|---|
| | WMO_INST_TYPE:_FillValue = " "; | |
| JULD | double JULD(N_PROF);<br>JULD:long_name = "Julian day (UTC) of the station relative to REFERENCE_DATE_TIME";<br>JULD:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD:conventions = "Relative julian days with decimal part (as parts of day)";<br>JULD:_FillValue = 999999.; | Julian day of the profile[1].<br>The integer part represents the day, the decimal part represents the time of the profile.<br>Date and time are in universal time coordinates.<br>The julian day is relative to REFERENCE_DATE_TIME.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_QC | char JULD_QC(N_PROF);<br>JULD_QC:long_name = "Quality on Date and Time";<br>JULD_QC:conventions = "Argo reference table 2";<br>JULD_QC:_FillValue = " "; | Quality flag on JULD date and time.<br>The flag scale is described in the reference table 2.<br>Example :<br>1 : the date and time seems correct. |
| JULD_LOCATION | double JULD_LOCATION(N_PROF);<br>JULD:long_name = "Julian day (UTC) of the location relative to REFERENCE_DATE_TIME ";<br>JULD:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD:conventions = "Relative julian days with decimal part (as parts of day)";<br>JULD:_FillValue = 999999.; | Julian day of the location of the profile (1).<br>The integer part represents the day, the decimal part represents the time of the profile.<br>Date and time are in universal time coordinates.<br>The julian day is relative to REFERENCE_DATE_TIME.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| LATITUDE | double LATITUDE(N_PROF);<br>LATITUDE:long_name = "Latitude of the station, best estimate";<br>LATITUDE:units = "degree_north";<br>LATITUDE:_FillValue = 99999.;<br>LATITUDE:valid_min = -90.;<br>LATITUDE:valid_max = 90.; | Latitude of the profile.<br>Unit : degree north<br>This field contains the best estimated latitude.<br>The latitude value may be improved in delayed mode.<br>The measured locations of the float are located in the trajectory file.<br>Example : 44.4991 : 44° 29' 56.76'' N |
| LONGITUDE | double LONGITUDE(N_PROF);<br>LONGITUDE:long_name = "Longitude of the station, best estimate";<br>LONGITUDE:units = "degree_east";<br>LONGITUDE:_FillValue = 99999.;<br>LONGITUDE:valid_min = -180.;<br>LONGITUDE:valid_max = 180.; | Longitude of the profile.<br>Unit : degree east<br>This field contains the best estimated longitude.<br>The longitude value may be improved in delayed mode.<br>The measured locations of the float are located in the trajectory file.<br>Example : 16.7222 : 16° 43' 19.92'' E |
| POSITION_QC | char POSITION_QC(N_PROF);<br>POSITION_QC:long_name = "Quality on position (latitude and longitude)";<br>POSITION_QC:conventions = "Argo reference table 2";<br>POSITION_QC:_FillValue = " "; | Quality flag on position.<br>The flag on position is set according to (LATITUDE, LONGITUDE) quality.<br>The flag scale is described in the reference table 2.<br>Example : 1 : position seems correct. |
| POSITIONING_SYSTEM | char POSITIONING_SYSTEM(N_PROF, STRING8);<br>POSITIONING_SYSTEM:long_name = "Positioning system";<br>POSITIONING_SYSTEM:_FillValue = " "; | Name of the system in charge of positioning the float locations from reference table 9.<br>Examples : ARGOS |
| PROFILE_<PARAM>_QC | char PROFILE_<PARAM>_QC(N_PROF);<br>PROFILE_<PARAM>_QC:long_name = "Global quality flag of <PARAM> profile";<br>PROFILE_<PARAM>_QC:conventions = "Argo reference table 2a";<br>PROFILE_<PARAM>_QC:_FillValue = " "; | Global quality flag on the PARAM profile.<br>PARAM is among the STATION_PARAMETERS.<br>The overall flag is set to indicate the percentage of good data in the profile as described in reference table 2a.<br>Example :<br>PROFILE_TEMP_QC = A : the temperature profile contains only good values |

[1] Assume that a float profiles on its ascent. When the float first comes to the surface, it begins to transmit data. Each data transmission has a time attached to it and the earliest time is what is recorded in JULD. It is possible that the first transmission from a float cannot be used to derive its location. In this case, the time of location, JULD_LOCATION, is different and later than the time of the profile.

| | | |
|---|---|---|
| | | PROFILE_PSAL_QC = C : the salinity profile contains 50% to 75% good values |

### 2.2.4. Measurements for each profile

This section contains information on each level of each profile.
Each variable in this section has a N_PROF (number of profiles), N_LEVELS (number of pressure levels) dimension.

The original data received from the float and examined by real-time quality control should be placed in the <PARAM> and the QC flags set by the real-time process should be placed in the <PARAM>_QC field. The values and flags in the <PARAM> fields should never be altered.

Each parameter can be adjusted (in delayed-mode, but also in real-time if appropriate). In that case, <PARAM>_ADJUSTED contains the adjusted values, <PARAM>_ADJUSTED_QC contains the QC flags set by the delayed-mode process, and <PARAM>_ADJUSTED_ERROR contains the adjustment uncertainties.

A real-time data file with no adjusted data has an adjusted section with fill values (<PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR).

The Argo profile delayed mode QC is described in "Argo quality control manual" by Annie Wong et Al.

| Name | Definition | Comment |
|---|---|---|
| <PARAM> | float <PARAM>(N_PROF, N_LEVELS);<br><PARAM>:long_name = "<X>";<br><PARAM>:_FillValue = <X>;<br><PARAM>:units = "<X>";<br><PARAM>:valid_min = <X>;<br><PARAM>:valid_max = <X>;<br><PARAM>:comment = "<X>";<br><PARAM>:C_format = "<X>";<br><PARAM>:FORTRAN_format = "<X>";<br><PARAM>:resolution = <X>; | <PARAM> contains the original values of a parameter listed in reference table 3.<br><X> : this field is specified in the reference table 3. |
| <PARAM>_QC | char <PARAM>_QC(N_PROF, N_LEVELS);<br><PARAM>_QC:long_name = "quality flag";<br><PARAM>_QC:conventions = "Argo reference table 2";<br><PARAM>_QC:_FillValue = " "; | Quality flag applied on each <PARAM> values.<br>The flag scale is specified in table 2. |
| <PARAM>_ADJUSTED | float <PARAM>_ADJUSTED(N_PROF, N_LEVELS);<br><PARAM>_ADJUSTED:long_name = "<X>";<br><PARAM>_ADJUSTED:_FillValue = <X>;<br><PARAM>_ADJUSTED:units = "<X>";<br><PARAM>_ADJUSTED:valid_min = <X>;<br><PARAM>_ADJUSTED:valid_max = <X>;<br><PARAM>_ADJUSTED:comment = "<X>";<br><PARAM>_ADJUSTED:C_format = "<X>";<br><PARAM>_ADJUSTED:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED:resolution= <X>; | <PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter.<br><X> : this field is specified in the reference table 3.<br><PARAM>_ADJUSTED is mandatory.<br>When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_ADJUSTED_QC | char <PARAM>_ADJUSTED_QC(N_PROF, N_LEVELS);<br><PARAM>_ADJUSTED_QC:long_name = "quality flag";<br><PARAM>_ADJUSTED_QC:conventions = "Argo reference table 2";<br><PARAM>_ADJUSTED_QC:_FillValue = " "; | Quality flag applied on each <PARAM>_ADJUSTED values.<br>The flag scale is specified in reference table 2.<br><PARAM>_ADJUSTED_QC is mandatory.<br>When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_ADJUSTED_ERROR | float <PARAM>_ADJUSTED_ERROR(N_PROF, N_LEVELS);<br><PARAM>_ADJUSTED_ERROR:long_name = "<X>";<br><PARAM>_ADJUSTED_ERROR:_FillValue = <X>;<br><PARAM>_ADJUSTED_ERROR:units = "<X>";<br><PARAM>_ADJUSTED_ERROR:comment = "Contains the | <PARAM>_ADJUSTED_ERROR contains the error on the adjusted values of the parameter.<br><X> : this field is specified in the reference table 3.<br><PARAM>_ADJUSTED_ERROR is |

| | | |
|---|---|---|
| | error on the adjusted values as determined by the delayed mode QC process.";<br><PARAM>_ADJUSTED_ERROR:C_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:resolution= <X>; | mandatory. When no adjustment is performed, the FillValue is inserted. |

*Example of a profiling float performing temperature measurements with adjusted values of temperature :*

| **Parameter definition : PRES, TEMP, TEMP_ADJUSTED** |
|---|
| float PRES(N_PROF, N_LEVELS);<br>PRES:long_name = "SEA PRESSURE (sea surface = 0)";<br>PRES:_FillValue = 99999.f;<br>PRES:units = "decibar";<br>PRES:valid_min = 0.f;<br>PRES:valid_max = 1200.f;<br>PRES:comment = "In situ measurement, sea surface = 0";<br>PRES:C_format = "7.1f" ;<br>PRES:FORTRAN_format= "F7.1";<br>PRES:resolution= 0.1f;<br><br>char PRES_QC(N_PROF, N_LEVELS);<br>PRES_QC:long_name = "quality flag";<br>PRES_QC:conventions = "Argo reference table 2";<br>PRES_QC:_FillValue = " ";<br><br>float TEMP(N_PROF, N_LEVELS);<br>TEMP:long_name = "SEA TEMPERATURE";<br>TEMP:_FillValue = 99999.f;<br>TEMP:units = "degree_Celsius";<br>TEMP:valid_min = -2.f;<br>TEMP:valid_max = 40.f;<br>TEMP:comment = "In situ measurement";<br>TEMP:C_format = "%9.3f";<br>TEMP:FORTRAN_format = "F9.3";<br>TEMP:resolution = 0.001f;<br><br>char TEMP_QC(N_PROF, N_LEVELS);<br>TEMP_QC:long_name = "quality flag";<br>TEMP_QC:conventions = "Argo reference table 2";<br>TEMP_QC:_FillValue = " ";<br><br>float TEMP_ADJUSTED(N_PROF, N_LEVELS);<br>TEMP_ADJUSTED:long_name = "ADJUSTED SEA TEMPERATURE";<br>TEMP_ADJUSTED:_FillValue = 99999.f;<br>TEMP_ADJUSTED:units = "degree_Celsius";<br>TEMP_ADJUSTED:valid_min = -2.f;<br>TEMP_ADJUSTED:valid_max = 40.f;<br>TEMP_ADJUSTED:comment = "Adjusted parameter";<br>TEMP_ADJUSTED:C_format = "%9.3f";<br>TEMP_ADJUSTED:FORTRAN_format= "F9.3";<br>TEMP_ADJUSTED:resolution= 0.001f;<br><br>char TEMP_ADJUSTED_QC(N_PROF, N_LEVELS);<br>TEMP_ADJUSTED QC:long_name = "quality flag";<br>TEMP_ADJUSTED QC:conventions = "Argo reference table 2";<br>TEMP_ADJUSTED_QC:_FillValue = " ";<br><br>float TEMP_ADJUSTED_ERROR(N_PROF, N_LEVELS);<br>TEMP_ADJUSTED_ERROR:long_name = "ERROR ON ADJUSTED SEA TEMPERATURE";<br>TEMP_ADJUSTED_ERROR:_FillValue = 99999.f;<br>TEMP_ADJUSTED_ERROR:units = "degree_Celsius";<br>TEMP_ADJUSTED_ERROR:comment = "Contains the error on the adjusted values as determined by the delayed mode QC process.";<br>TEMP_ADJUSTED_ERROR :C_format = "%9.3f";<br>TEMP_ADJUSTED_ERROR :FORTRAN_format= "F9.3";<br>TEMP_ADJUSTED_ERROR:resolution= 0.001f; |

### 2.2.5. Calibration information for each profile

Calibrations are applied to parameters to create adjusted parameters. Different calibration methods will be used by groups processing Argo data. When a method is applied, its description is stored in the following fields.

This section contains calibration information for each parameter of each profile.

Each item of this section has a N_PROF (number of profiles), N_CALIB (number of calibrations), N_PARAM (number of parameters) dimension.

If no calibration is available, N_CALIB is set to 1, all values of calibration section are set to fill values.

| Name | Definition | Comment |
|---|---|---|
| PARAMETER | char PARAMETER(N_PROF, N_CALIB, N_PARAM,STRING16);<br>PARAMETER:long_name = "List of parameters with calibration information";<br>PARAMETER:conventions = "Argo reference table 3";<br>PARAMETER:_FillValue = " "; | Name of the calibrated parameter. The list of parameters is in reference table 3.<br>Example : PSAL |
| SCIENTIFIC_CALIB_EQUATION | Char SCIENTIFIC_CALIB_EQUATION(N_PROF, N_CALIB, N_PARAM,STRING256);<br>SCIENTIFIC_CALIB_EQUATION:long_name = "Calibration equation for this parameter";<br>SCIENTIFIC_CALIB_EQUATION:_FillValue = " "; | Calibration equation applied to the parameter.<br>Example :<br>Tc = a1 * T + a0 |
| SCIENTIFIC_CALIB_COEFFICIENT | Char SCIENTIFIC_CALIB_COEFFICIENT(N_PROF, N_CALIB, N_PARAM,STRING256);<br>SCIENTIFIC_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation";<br>SCIENTIFIC_CALIB_COEFFICIENT:_FillValue = " "; | Calibration coefficients for this equation.<br>Example :<br>a1=0.99997 , a0=0.0021 |
| SCIENTIFIC_CALIB_COMMENT | Char SCIENTIFIC_CALIB_COMMENT(N_PROF, N_CALIB, N_PARAM,STRING256);<br>SCIENTIFIC_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration";<br>SCIENTIFIC_CALIB_COMMENT:_FillValue = " "; | Comment about this calibration<br>Example :<br>The sensor is not stable |
| CALIBRATION_DATE | Char CALIBRATION_DATE (N_PROF N_CALIB, N_PARAM, DATE_TIME)<br>CALIBRATION_DATE:_FillValue = " "; | Date of the calibration.<br>Example : 20011217161700 |

### 2.2.6. History information for each profile

This section contains history information for each action performed on each profile by a data centre.

Each item of this section has a N_HISTORY (number of history records), N_PROF (number of profiles) dimension.

A history record is created whenever an action is performed on a profile.

The recorded actions are coded and described in the history code table from the reference table 7.

On the GDAC, multi-profile history section is empty to reduce the size of the file. History section is available on mono-profile files, or in multi-profile files distributed from the web data selection.

| Name | Definition | Comment |
|---|---|---|
| HISTORY_INSTITUTION | char HISTORY_INSTITUTION ( N_HISTORY, N_PROF, STRING4);<br>HISTORY_INSTITUTION:long_name = "Institution which performed action";<br>HISTORY_INSTITUTION:conventions = "Argo reference table 4";<br>HISTORY_INSTITUTION:_FillValue = " "; | Institution that performed the action.<br>Institution codes are described in reference table 4.<br>Example : ME for MEDS |
| HISTORY_STEP | char HISTORY_STEP ( N_HISTORY, N_PROF, STRING4);<br>HISTORY_STEP:long_name = "Step in data processing";<br>HISTORY_STEP:conventions = "Argo reference table 12";<br>HISTORY_STEP:_FillValue = " "; | Code of the step in data processing for this history record. The step codes are described in reference table 12.<br>Example :<br>ARGQ : Automatic QC of data reported in real-time has been performed |
| HISTORY_SOFTWARE | Char HISTORY_SOFTWARE ( N_HISTORY, N_PROF, STRING4);<br>HISTORY_SOFTWARE:long_name = "Name of software which performed action";<br>HISTORY_SOFTWARE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE:_FillValue = " "; | Name of the software that performed the action.<br>This code is institution dependent.<br>Example : WJO |
| HISTORY_SOFTWARE_RELEASE | Char HISTORY_SOFTWARE_RELEASE ( N_HISTORY, N_PROF, STRING4);<br>HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action";<br>HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE_RELEASE:_FillValue = " "; | Version of the software.<br>This name is institution dependent.<br>Example : «1.0» |
| HISTORY_REFERENCE | char HISTORY_REFERENCE ( N_HISTORY, N_PROF, STRING64);<br>HISTORY_REFERENCE:long_name = "Reference of database";<br>HISTORY_REFERENCE:conventions = "Institution dependent";<br>HISTORY_REFERENCE:_FillValue = " "; | Code of the reference database used for quality control in conjunction with the software.<br>This code is institution dependent.<br>Example : WOD2001 |
| HISTORY_DATE | char HISTORY_DATE( N_HISTORY, N_PROF, DATE_TIME);<br>HISTORY_DATE:long_name = "Date the history record was created";<br>HISTORY_DATE:conventions = "YYYYMMDDHHMISS"; | Date of the action.<br>Example : 20011217160057 |

| | | |
|---|---|---|
| | HISTORY_DATE:_FillValue = " "; | |
| HISTORY_ACTION | char HISTORY_ACTION( N_HISTORY, N_PROF, STRING4);<br>HISTORY_ACTION:long_name = "Action performed on data";<br>HISTORY_ACTION:conventions = "Argo reference table 7";<br>HISTORY_ACTION:_FillValue = " "; | Name of the action.<br>The action codes are described in reference table 7.<br>Example : QCF$ for QC failed |
| HISTORY_PARAMETER | char HISTORY_PARAMETER( N_HISTORY, N_PROF, STRING16);<br>HISTORY_PARAMETER:long_name = "Station parameter action is performed on";<br>HISTORY_PARAMETER:conventions = "Argo reference table 3";<br>HISTORY_PARAMETER:_FillValue = " "; | Name of the parameter on which the action is performed.<br>Example : PSAL |
| HISTORY_START_PRES | float HISTORY_START_PRES( N_HISTORY, N_PROF);<br>HISTORY_START_PRES:long_name = "Start pressure action applied on";<br>HISTORY_START_PRES:_FillValue = 99999.f;<br>HISTORY_START_PRES:units = "decibar"; | Start pressure the action is applied to.<br>Example : 1500.0 |
| HISTORY_STOP_PRES | float HISTORY_STOP_PRES( N_HISTORY, N_PROF);<br>HISTORY_STOP_PRES:long_name = "Stop pressure action applied on";<br>HISTORY_STOP_PRES:_FillValue = 99999.f;<br>HISTORY_STOP_PRES:units = "decibar"; | Stop pressure the action is applied to. This should be greater than START_PRES.<br>Example : 1757.0 |
| HISTORY_PREVIOUS_VALUE | float HISTORY_PREVIOUS_VALUE( N_HISTORY, N_PROF);<br>HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action";<br>HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f; | Parameter or flag of the previous value before action.<br>Example : 2 (probably good) for a flag that was changed to 1 (good) |
| HISTORY_QCTEST | char HISTORY_QCTEST( N_HISTORY, N_PROF, STRING16);<br>HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)";<br>HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$; tests failed when ACTION=QCF$";<br>HISTORY_QCTEST:_FillValue = " "; | This field records the tests performed when ACTION is set to QCP$ (qc performed), the test failed when ACTION is set to QCF$ (qc failed).<br>The QCTEST codes are describe in reference table 11.<br><br>Example : 0A (in hexadecimal form) |

The usage of the History section is described in §5 "Using the History section of the Argo netCDF Structure".

## 2.3. Trajectory format

An Argo trajectory file contains all received locations of an Argo float. There is one trajectory file per float.

In addition to locations, a trajectory file may contain measurements such as temperature, salinity or conductivity performed at some or all locations.

For file naming conventions, see §4.1 .

### 2.3.1. Dimensions and definitions

| Name | Definition | Comment |
|---|---|---|
| DATE_TIME | DATE_TIME = 14; | This dimension is the length of an ASCII date and time value.<br>Date_time convention is : YYYYMMDDHHMISS<br>YYYY : year<br>MM : month<br>DD : day<br>HH : hour of the day<br>MI : minutes<br>SS : seconds<br>Date and time values are always in universal time coordinates (UTC).<br>Examples :<br>20010105172834 : January 5th 2001 17:28:34<br>19971217000000 : December 17th 1997 00:00:00 |
| STRING256<br>STRING64<br>STRING32<br>STRING16<br>STRING8<br>STRING4<br>STRING2 | STRING256 = 256;<br>STRING64 = 64;<br>STRING32 = 32;<br>STRING16 = 16;<br>STRING8 = 8;<br>STRING4 = 4;<br>STRING2 = 2; | String dimensions from 2 to 256. |
| N_PARAM | N_PARAM = <int value> ; | Maximum number of parameters measured or calculated for a pressure sample.<br>Examples :<br>(pressure, temperature) : N_PARAM = 2<br>(pressure, temperature, salinity) : N_PARAM = 3<br>(pressure, temperature, conductivity, salinity) : N_PARAM = 4 |
| N_MEASUREMENT | N_MEASUREMENT = unlimited; | This dimension is the number of recorded locations and measurements of the file. |
| N_CYCLE | N_CYCLE = <int value> ; | Maximum number of cycles performed by the float.<br>This dimension depends on the data set.<br>Example : N_CYCLE = 100 |
| N_HISTORY | N_HISTORY = <int value> ; | Maximum number of history records for a location. This dimension depends on the data set<br>Exemple : N_HISTORY = 10 |
| N_HISTORY2 | N_HISTORY2 = <int value> ; | Maximum dimension of the history record with respect to the time axis. This dimension depends on the data set.<br>Example : N_HISTORY2 = 2 if only two values were corrected |

### 2.3.2. General information on the trajectory file

This section contains information about the whole file.

| Name | Definition | Comment |
|---|---|---|
| DATA_TYPE | char DATA_TYPE(STRING16);<br>DATA_TYPE:comment = "Data type";<br>DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file.<br>The list of acceptable data types is in the reference table 1.<br>Example : Argo trajectory |
| FORMAT_VERSION | char FORMAT_VERSION(STRING4);<br>FORMAT_VERSION:comment = "File format version ";<br>FORMAT_VERSION:_FillValue = " "; | File format version<br>Example : «2.2» |
| HANDBOOK_VERSION | char HANDBOOK_VERSION(STRING4);<br>HANDBOOK_VERSION:comment = "Data handbook version";<br>HANDBOOK_VERSION:_FillValue = " "; | Version number of the data handbook.<br>This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook.<br>Example : «1.0» |
| REFERENCE_DATE_TIME | char REFERENCE_DATE_TIME(DATE_TIME);<br>REFERENCE_DATE_TIME:comment = "Date of reference for Julian days";<br>REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMISS";<br>REFERENCE_DATE_TIME:_FillValue = " "; | Date of reference for julian days.<br>The recommended reference date time is<br>«19500101000000» : January 1$^{st}$ 1950 00:00:00 |

### 2.3.3. General information on the float

This section contains general information on the float.

| Name | Definition | Comment |
|---|---|---|
| PLATFORM_NUMBER | char PLATFORM_NUMBER(STRING8);<br>PLATFORM_NUMBER:long_name = "Float unique identifier";<br>PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII";<br>PLATFORM_NUMBER:_FillValue = " "; | WMO float identifier.<br>WMO is the World Meteorological Organization.<br>This platform number is unique.<br>Example : 6900045 |
| PROJECT_NAME | char PROJECT_NAME(STRING64);<br>PROJECT_NAME:comment = "Name of the project";<br>PROJECT_NAME:_FillValue = " "; | Name of the project which operates the float that performed the trajectory.<br>Example : GYROSCOPE (EU project for ARGO program) |
| PI_NAME | char PI_NAME (STRING64);<br>PI_NAME:comment = "Name of the principal investigator";<br>PI_NAME:_FillValue = " "; | Name of the principal investigator in charge of the float.<br>Example : Yves Desaubies |
| TRAJECTORY_PARAMETERS | char TRAJECTORY_PARAMETERS(N_PARAM,STRING16);<br>TRAJECTORY_PARAMETERS:long_name = "List of available parameters for the station";<br>TRAJECTORY_PARAMETERS:conventions = "Argo reference table 3";<br>TRAJECTORY_PARAMETERS:_FillValue = " "; | List of parameters contained in this trajectory file.<br>The parameter names are listed in reference table 3.<br>Examples : TEMP, PSAL, CNDC<br>TEMP : temperature<br>PSAL : practical salinity<br>CNDC : conductvity |
| DATA_CENTRE | char DATA_CENTRE(STRING2);<br>DATA_CENTRE:long_name = "Data centre in charge of float data processing";<br>DATA_CENTRE:conventions = "Argo reference table 4";<br>DATA_CENTRE:_FillValue = " "; | Code for the data centre in charge of the float data management.<br>The data centre codes are described in the reference table 4.<br>Example : ME for MEDS |
| DATE_CREATION | char DATE_CREATION(DATE_TIME);<br>DATE_CREATION:comment = "Date of | Date and time (UTC) of creation of this file.<br>Format : YYYYMMDDHHMISS |

| | | |
|---|---|---|
| | file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " "; | Example : 20011229161700 : December 29th 2001 16 :17 :00 |
| DATE_UPDATE | char DATE_UPDATE(DATE_TIME); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 09 :05 :00 |
| DATA_STATE_INDICATOR | char DATA_STATE_INDICATOR(STRING4); DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through"; DATA_STATE_INDICATOR:conventions = "Argo reference table 6"; DATA_STATE_INDICATOR:_FillValue = " "; | Degree of processing the data has passed through. The data state indicator is described in the reference table 6. |
| INST_REFERENCE | char INST_REFERENCE(STRING64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number"; INST_REFERENCE:_FillValue = " "; | Information about instrument : brand, type, serial number Example : APEX-SBE 259 |
| WMO_INST_TYPE | char WMO_INST_TYPE(STRING4); WMO_INST_TYPE:long_name = "Coded instrument type"; WMO_INST_TYPE:conventions = "Argo reference table 8"; WMO_INST_TYPE:_FillValue = " "; | Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 8. Example : 831 |
| POSITIONING_SYSTEM | char POSITIONING_SYSTEM(STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:_FillValue = " "; | Name of the system used to derive the float locations, see reference table 9. Examples : ARGOS |

### 2.3.4. Locations and measurements from the float

This section contains locations for one Argo float. It may also contain measurements performed along the trajectory.

Each field in this section has a N_MEASUREMENT dimension.

N_MEASUREMENT is the number of locations (or measurement) received from the float.

When no parameter is measured along the trajectory, N_PARAM (number of parameters) and any field with a N_PARAM dimension are removed from the file : PARAM, PARAM_QC, PARAM_ADJUSTED, PARAM_ADJUSTED_QC, PARAM_ADJUSTED_ERROR and TRAJECTORY_PARAMETERS.

The original data received from the float and examined by real-time quality control should be placed in the <PARAM> and the QC flags set by the real-time process should be placed in the <PARAM>_QC field. The values and flags in the <PARAM> fields should never be altered.

Each parameter can be adjusted. In that case, <PARAM>_ADJUSTED contains the adjusted values, <PARAM>_ADJUSTED_QC contains the QC flags set by the delayed-mode process, and <PARAM>_ADJUSTED_ERROR contains the adjustment uncertainties.

A file with no adjusted data contains adjusted sections with fill values (<PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR).

| Name | Definition | Comment |
|---|---|---|
| DATA_MODE | char DATA_MODE(N_MEASUREMENT);<br>DATA_MODE:long_name = "Delayed mode or real time data";<br>DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment";<br>DATA_MODE:_FillValue = " "; | Indicates if the profile contains real time or delayed mode data.<br>R : real time data<br>D : delayed mode data<br>A : real time data with adjusted values |
| DC_REFERENCE | char DC_REFERENCE(N_MEASUREMENT STRING32);<br>DC_REFERENCE:long_name = "Location unique identifier in data centre";<br>DC_REFERENCE:conventions = "Data centre convention";<br>DC_REFERENCE:_FillValue = " "; | Unique identifier of the location in the data centre.<br>Data centres may have different identifier schemes.<br>DC_REFERENCE is therefore not always unique across data centres. |
| JULD | double JULD(N_MEASUREMENT);<br>JULD:long_name = "Julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME";<br>JULD:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD:conventions = "Relative julian days with decimal part (as parts of the day)";<br>JULD:_FillValue = 999999.; | Julian day of the location (or measurement).<br>The integer part represents the day, the decimal part represents the time of the measurement.<br>Date and time are in universal time coordinates.<br>The julian day is relative to REFERENCE_DATE_TIME.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_QC | char JULD_QC(N_MEASUREMENT);<br>JULD_QC:long_name = "Quality on date and time";<br>JULD_QC:conventions = "Argo reference table 2";<br>JULD_QC:_FillValue = " "; | Quality flag on JULD date and time.<br>The flag scale is described in the reference table 2.<br>Example :<br>1 : the date and time seems correct. |
| LATITUDE | double LATITUDE(N_MEASUREMENT);<br>LATITUDE:long_name = "Latitude of each location"; | Latitude of the location (or measurement).<br>Unit : degree north<br>Example : 44.4991 for 44° 29' 56.76'' N |

| | | |
|---|---|---|
| | LATITUDE:units = "degree_north";<br>LATITUDE:_FillValue = 99999.;<br>LATITUDE:valid_min = -90.;<br>LATITUDE:valid_max = 90.; | |
| LONGITUDE | double LONGITUDE(N_MEASUREMENT);<br>LONGITUDE:long_name = "Longitude of each location";<br>LONGITUDE:units = "degree_east";<br>LONGITUDE:_FillValue = 99999.;<br>LONGITUDE:valid_min = -180.;<br>LONGITUDE:valid_max = 180.; | Longitude of the location (or measurement).<br>Unit : degree east<br>Example : 16.7222 for 16° 43' 19.92'' E |
| POSITION_ACCURACY | char POSITION_ACCURACY(N_MEASUREMENT);<br>POSITION_ACCURACY:long_name = "Estimated accuracy in latitude and longitude";<br>POSITION_ACCURACY:conventions = "Argo reference table 5";<br>POSITION_ACCURACY:_FillValue = " "; | Position accuracy received from the positioning system.<br>The location classes from ARGOS are described in the reference table 5.<br>Example : 3 for a latitude and longitude accuracy < 150 m. |
| POSITION_QC | char POSITION_QC(N_MEASUREMENT);<br>POSITION_QC:long_name = "Quality on position";<br>POSITION_QC:conventions = "Argo reference table 2";<br>POSITION_QC:_FillValue = " "; | Quality flag on position.<br>The flag on position is set according to (LATITUDE, LONGITUDE, JULD) quality.<br>The flag scale is described in the reference table 2.<br>Example : 1 : position seems correct. |
| CYCLE_NUMBER | int CYCLE_NUMBER(N_MEASUREMENT);<br>CYCLE_NUMBER:long_name = "Float cycle number of the measurement";<br>CYCLE_NUMBER:conventions = "0..N, 0 : launch cycle, 1 : first complete cycle";<br>CYCLE_NUMBER:_FillValue = 99999; | Cycle number of the float for this measurement.<br>For one cycle number, there are usually several locations/measurement received.<br>Example : 17 for measurements performed during the 17th cycle of the float. |
| <PARAM> | float <PARAM>(N_MEASUREMENT);<br><PARAM>:long_name = "<X>";<br><PARAM>:_FillValue = <X>;<br><PARAM>:units = "<X>";<br><PARAM>:valid_min = <X>;<br><PARAM>:valid_max = <X>;<br><PARAM>:comment = "<X>";<br><PARAM>:C_format = "<X>";<br><PARAM>:FORTRAN_format = "<X>";<br><PARAM>:resolution = <X>; | <PARAM> contains the original values of a parameter listed in reference table 3.<br><X> : this field is specified in the reference table 3. |
| <PARAM>_QC | char <PARAM>_QC(N_MEASUREMENT);<br><PARAM>_QC:long_name = "quality flag";<br><PARAM>_QC:conventions = "Argo reference table 2";<br><PARAM>_QC:_FillValue = " "; | Quality flag applied on each <PARAM> values.<br>The flag scale is specified in table 2. |
| <PARAM>_ADJUSTED | float <PARAM>_ADJUSTED(N_MEASUREMENT);<br><PARAM>_ADJUSTED:long_name = "<X>";<br><PARAM>_ADJUSTED:_FillValue = <X>;<br><PARAM>_ADJUSTED:units = "<X>";<br><PARAM>_ADJUSTED:valid_min = <X>;<br><PARAM>_ADJUSTED:valid_max = <X>;<br><PARAM>_ADJUSTED:comment = "<X>";<br><PARAM>_ADJUSTED:C_format = "<X>";<br><PARAM>_ADJUSTED:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED:resolution= <X>; | <PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter.<br><X> : this field is specified in the reference table 3.<br><PARAM>_ADJUSTED is mandatory. When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_ADJUSTED_QC | char <PARAM>_ADJUSTED_QC(N_MEASUREMENT);<br><PARAM>_ADJUSTED_QC:long_name = "quality flag";<br><PARAM>_ADJUSTED_QC:conventions = "Argo reference table 2";<br><PARAM>_ADJUSTED_QC:_FillValue = " "; | Quality flag applied on each <PARAM>_ADJUSTED values.<br>The flag scale is specified in reference table 2.<br><PARAM>_ADJUSTED_QC is mandatory. When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_ADJUSTED_ERROR | float <PARAM>_ADJUSTED_ERROR(N_MEASUREMENT);<br><PARAM>_ADJUSTED_ERROR:long_name = "<X>";<br><PARAM>_ADJUSTED_ERROR:_FillValue = <X>; | <PARAM>_ADJUSTED_ERROR contains the error on the adjusted values of the parameter.<br><X> : this field is specified in the reference table 3.<br><PARAM>_ADJUSTED_ERROR is mandatory. When no adjustment is performed, the FillValue is inserted. |

| | | |
|---|---|---|
| | <PARAM>_ADJUSTED_ERROR:units = "<X>";<br><PARAM>_ADJUSTED_ERROR:comment = "Contains the error on the adjusted values as determined by the delayed mode QC process.";<br><PARAM>_ADJUSTED_ERROR:C_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:resolution= <X>; | |

## 2.3.5. Cycle information from the float

This section contains information on the cycles performed by the float.

Each field in this section has a N_CYCLE dimension.

N_CYCLE is the number of cycles performed by the float.

| Name | Definition | Comment |
|---|---|---|
| JULD_ASCENT_START | double JULD_ASCENT_START(N_CYCLE);<br>JULD_ASCENT_START:long_name = "Start date of the ascending profile";<br>JULD_ASCENT_START:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD_ASCENT_START:conventions = "Relative julian days with decimal part (as part of day)";<br>JULD_ASCENT_START:_FillValue=999999.; | Julian day (UTC) of the beginning of the ascending profile.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_ASCENT_START_STATUS | Char JULD_ASCENT_START_STATUS(N_CYCLE);<br>JULD_ASCENT_START_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";<br>JULD_ASCENT_START_STATUS:_FillValue = " "; | 0 : date comes from the float meta data<br>1 : date is estimated<br>2 : date is transmitted by the float<br>9 : date is unknown |
| JULD_ASCENT_END | double JULD_ASCENT_END(N_CYCLE);<br>JULD_ASCENT_END:long_name = "End date of the ascending profile";<br>JULD_ASCENT_END:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD_ASCENT_END:conventions = "Relative julian days with decimal part (as part of day)";<br>JULD_ASCENT_END:_FillValue=999999.; | Julian day (UTC) of the end of the ascending profile.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_ASCENT_END_STATUS | Char JULD_ASCENT_END_STATUS(N_CYCLE);<br>JULD_ASCENT_END_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";<br>JULD_ASCENT_END_STATUS:_FillValue = " "; | 0 : date comes from the float meta data<br>1 : date is estimated<br>2 : date is transmitted by the float<br>9 : date is unknown |
| JULD_DESCENT_START | double JULD_DESCENT_START(N_CYCLE);<br>JULD_DESCENT_START:long_name = "Descent start date of the cycle";<br>JULD_DESCENT_START:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD_DESCENT_START:conventions = "Relative julian days with decimal part (as part of day)";<br>JULD_DESCENT_START:_FillValue=99999 | Julian day (UTC) of the beginning of the descending profile.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |

| | | |
|---|---|---|
| | 9.; | |
| JULD_DESCENT_START_STATUS | Char JULD_DESCENT_START_STATUS(N_CYCLE); JULD_DESCENT_START_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted"; JULD_DESCENT_START_STATUS:_FillValue = " "; | 0 : date comes from the float meta data<br>1 : date is estimated<br>2 : date is transmitted by the float<br>9 : date is unknown |
| JULD_DESCENT_END | double JULD_DESCENT_END(N_CYCLE); JULD_DESCENT_END:long_name = "Descent end date of the cycle"; JULD_DESCENT_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DESCENT_END:conventions = "Relative julian days with decimal part (as part of day) "; JULD_DESCENT_END:_FillValue=999999.; | Julian day (UTC) of the end of the descending profile.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_DESCENT_END_STATUS | char JULD_DESCENT_END_STATUS(N_CYCLE); JULD_DESCENT_END_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted"; JULD_DESCENT_END_STATUS:_FillValue = " "; | 0 : date comes from the float meta data<br>1 : date is estimated<br>2 : date is transmitted by the float<br>9 : date is unknown |
| JULD_START_TRANSMISSION | double JULD_START_TRANSMISSION(N_CYCLE); JULD_START_TRANSMISSION:long_name = "Start date of transmssion"; JULD_START_TRANSMISSION:units = "days since 1950-01-01 00:00:00 UTC"; JULD_START_TRANSMISSION:conventions = "Relative julian days with decimal part (as part of day)"; JULD_START_TRANSMISSION:_FillValue=999999.; | Julian day (UTC) of the beginning of data transmission.<br>Example :<br>18833.8013889885 : July 25 2001 19:14:00 |
| JULD_START_TRANSMISSION_STATUS | char JULD_START_TRANSMISSION_STATUS(N_CYCLE); JULD_START_TRANSMISSION_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted"; JULD_START_TRANSMISSION_STATUS:_FillValue = " "; | 0 : date comes from the float meta data<br>1 : date is estimated<br>2 : date is transmitted by the float<br>9 : date is unknown |
| GROUNDED | char GROUNDED(N_CYCLE); GROUNDED:long_name = "Did the profiler touch the ground for that cycle”; GROUNDED:conventions = "Y,N,U"; GROUNDED:_FillValue = " "; | GROUNDED indicates if the float touched the ground for that cycle.<br>Format : Y, N, U<br>Examples :<br>Y : yes the float touched the ground<br>N : no<br>U : unknown |

### 2.3.6. History information

This section contains history information for each action performed on each measurement.

Each item of this section has a N_MEASUREMENT (number of locations or measurements), N_HISTORY (number of history records) dimension.

| Name | Definition | Comment |
|---|---|---|
| HISTORY_INSTITUTION | char HISTORY_INSTITUTION (N_HISTORY, STRING4);<br>HISTORY_INSTITUTION:long_name = "Institution which performed action";<br>HISTORY_INSTITUTION:conventions = "Argo reference table 4";<br>HISTORY_INSTITUTION:_FillValue = " "; | Institution that performed the action.<br>Institution codes are described in reference table 4.<br>Example : ME for MEDS |
| HISTORY_STEP | char HISTORY_STEP (N_HISTORY, STRING4);<br>HISTORY_STEP:long_name = "Step in data processing";<br>HISTORY_STEP:conventions = "Argo reference table 12";<br>HISTORY_STEP:_FillValue = " "; | Code of the step in data processing for this history record. The step codes are described in reference table 12.<br>Example :<br>ARGQ : Automatic QC of data reported in real-time has been performed |
| HISTORY_SOFTWARE | Char HISTORY_SOFTWARE (N_HISTORY, STRING4);<br>HISTORY_SOFTWARE:long_name = "Name of software which performed action";<br>HISTORY_SOFTWARE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE:_FillValue = " "; | Name of the software that performed the action.<br>This code is institution dependent.<br>Example : WJO |
| HISTORY_SOFTWARE_RELEASE | Char HISTORY_SOFTWARE_RELEASE (N_HISTORY, STRING4);<br>HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action";<br>HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE_RELEASE:_FillValue = " "; | Version of the software.<br>This name is institution dependent.<br>Example : «1.0» |
| HISTORY_REFERENCE | char HISTORY_REFERENCE (N_HISTORY, STRING64);<br>HISTORY_REFERENCE:long_name = "Reference of database";<br>HISTORY_REFERENCE:conventions = "Institution dependent";<br>HISTORY_REFERENCE:_FillValue = " "; | Code of the reference database used for quality control in conjunction with the software.<br>This code is institution dependent.<br>Example : WOD2001 |
| HISTORY_DATE | char HISTORY_DATE(N_HISTORY, DATE_TIME);<br>HISTORY_DATE:long_name = "Date the history record was created";<br>HISTORY_DATE:conventions = "YYYYMMDDHHMISS";<br>HISTORY_DATE:_FillValue = " "; | Date of the action.<br>Example : 20011217160057 |
| HISTORY_ACTION | char HISTORY_ACTION (N_HISTORY, STRING64);<br>HISTORY_ACTION:long_name = "Action performed on data";<br>HISTORY_ACTION:conventions = "Argo reference table 7";<br>HISTORY_ACTION:_FillValue = " "; | Name of the action.<br>The action codes are described in reference table 7.<br>Example : QCF$ for QC failed |

| | | |
|---|---|---|
| HISTORY_PARAMETER | Char HISTORY_PARAMETER(N_HISTORY, STRING16);<br>HISTORY_PARAMETER:long_name = "Station parameter action is performed on";<br>HISTORY_PARAMETER:conventions = "Argo reference table 3";<br>HISTORY_PARAMETER:_FillValue = " "; | Name of the parameter on which the action is performed.<br>Example : PSAL |
| HISTORY_PREVIOUS_VALUE | Float HISTORY_PREVIOUS_VALUE(N_HISTORY, N_HISTORY2);<br>HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action";<br>HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f; | Parameter or flag of the previous value before action.<br>Example : 2 (probably good) for a flag that was changed to 1 (good) |
| HISTORY_INDEX_DIMENSION | char HISTORY_INDEX_DIMENSION(N_HISTORY); | Name of dimension to which HISTORY_START_INDEX and HISORY_STOP_INDEX Correspond.<br>C: N_CYCLE<br>M: N_MEASUREMENT |
| HISTORY_START_INDEX | int HISTORY_ START_INDEX (N_HISTORY);<br>HISTORY_START_INDEX:long_name = "Start index action applied on";<br>HISTORY_START_INDEX:_FillValue = 99999; | Start index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter<br>Example : 100 |
| HISTORY_STOP_INDEX | int HISTORY_ STOP_INDEX (N_HISTORY);<br>HISTORY_STOP_INDEX:long_name = "Stop index action applied on";<br>HISTORY_STOP_INDEX:_FillValue = 99999; | Stopt index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter<br>Example : 150 |
| HISTORY_QCTEST | char HISTORY_QCTEST(N_HISTORY, N_HISTORY2, STRING16);<br>HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)";<br>HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$; tests failed when ACTION=QCF$";<br>HISTORY_QCTEST:_FillValue = " "; | This field records the tests performed when ACTION is set to QCP$ (qc performed), the test failed when ACTION is set to QCF$ (qc failed).<br>The QCTEST codes are describe in reference table 11.<br><br>Example : 0A (in hexadecimal form) |

The usage of history section is described in §5 "Using the History section of the Argo netCDF Structure".

## 2.4. Meta-data format

An Argo meta-data file contains information about an Argo float.

For file naming conventions, see §4.1 .

### 2.4.1. Dimensions and definitions

| Name | Definition | Comment |
|---|---|---|
| DATE_TIME | DATE_TIME = 14; | This dimension is the length of an ASCII date and time value.<br>Date_time convention is : YYYYMMDDHHMISS<br>YYYY : year<br>MM : month<br>DD : day<br>HH : hour of the day<br>MI : minutes<br>SS : seconds<br>Date and time values are always in universal time coordinates (UTC).<br>Examples :<br>20010105172834 : January 5th 2001 17:28:34<br>19971217000000 : December 17th 1997 00:00:00 |
| STRING256<br>STRING64<br>STRING32<br>STRING16<br>STRING8<br>STRING4<br>STRING2 | STRING256 = 256;<br>STRING64 = 64;<br>STRING32 = 32;<br>STRING16 = 16;<br>STRING8 = 8;<br>STRING4 = 4;<br>STRING2 = 2; | String dimensions from 2 to 256. |
| N_CYCLES | N_CYCLES = <int value> ; | Number of different nominal cycles.<br>This value is usually set to 1 : all the cycles are programmed to be the same.<br>However, some floats may perform cycles with different programming.<br>Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5th cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2.<br>N_CYCLES = 2<br>The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1. |
| N_PARAM | N_PARAM=<int value> ; | Number of parameters measured or calculated for a pressure sample.<br>Examples :<br>(pressure, temperature) : N_PARAM = 2<br>(pressure, temperature, salinity) : N_PARAM = 3<br>(pressure, temperature, conductivity, salinity) : N_PARAM = 4 |

### 2.4.2. General information on the meta-data file

This section contains information about the whole file.

| Name | Definition | Comment |
|---|---|---|
| DATA_TYPE | char DATA_TYPE(STRING16);<br>DATA_TYPE:comment = "Data type";<br>DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file.<br>The list of acceptable data types is in the reference table 1.<br>Example : Argo meta-data |
| FORMAT_VERSION | char FORMAT_VERSION(STRING4);<br>FORMAT_VERSION:comment = "File format version ";<br>FORMAT_VERSION:_FillValue = " "; | File format version<br>Example : «2.2» |
| HANDBOOK_VERSION | char HANDBOOK_VERSION(STRING4);<br>HANDBOOK_VERSION:comment = "Data handbook version";<br>HANDBOOK_VERSION:_FillValue = " "; | Version number of the data handbook.<br>This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook.<br>Example : «1.0» |
| DATE_CREATION | char DATE_CREATION(DATE_TIME);<br>DATE_CREATION:comment = "Date of file creation ";<br>DATE_CREATION:conventions = "YYYYMMDDHHMISS";<br>DATE_CREATION:_FillValue = " "; | Date and time (UTC) of creation of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011229161700 : December 29th 2001 16:17:00 |
| DATE_UPDATE | char DATE_UPDATE(DATE_TIME);<br>DATE_UPDATE:long_name = "Date of update of this file";<br>DATE_UPDATE:conventions = "YYYYMMDDHHMISS";<br>DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 09:05:00 |

## 2.4.3. Float characteristics

This section contains the main characteristics of the float.

| Name | Definition | Comment |
|---|---|---|
| PLATFORM_NUMBER | char PLATFORM_NUMBER(STRING8);<br>PLATFORM_NUMBER:long_name = "Float unique identifier";<br>PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII";<br>PLATFORM_NUMBER:_FillValue = " "; | WMO float identifier.<br>WMO is the World Meteorological Organization.<br>This platform number is unique.<br>Example : 6900045 |
| PTT | char PTT (STRING256);<br>PTT:long_name = "Transmission identifier (ARGOS, ORBCOMM, etc.)";<br>PTT:_FillValue = " "; | Transmission identifier of the float.<br>Comma separated list for multi-beacon transmission.<br>Example :<br>22507 : the float is equipped with one ARGOS beacon.<br>22598,22768 : the float is equipped with 2 ARGOS beacons. |
| TRANS_SYSTEM | char TRANS_SYSTEM(STRING16);<br>TRANS_SYSTEM:long_name = "The telecommunications system used";<br>TRANS_SYSTEM:_FillValue = " "; | Name of the telecommunication system from reference table 10.<br>Example : ARGOS |
| TRANS_SYSTEM_ID | char TRANS_SYSTEM_ID(STRING32);<br>TRANS_SYSTEM_ID:long_name = "The program identifier used by the transmission system";<br>TRANS_SYSTEM_ID:_FillValue = " "; | Program identifier of the telecommunication subscription.<br>Example :<br>38511 is a program number for all the beacons of an ARGOS customer. |
| TRANS_FREQUENCY | char TRANS_FREQUENCY(STRING16);<br>TRANS_FREQUENCY:long_name = "The frequency of transmission from the float";<br>TRANS_FREQUENCY:units = "hertz";<br>TRANS_FREQUENCY:_FillValue = " "; | Frequency of transmission from the float.<br>Unit : hertz<br>Example : 1/44 |
| TRANS_REPETITION | float TRANS_REPETITION;<br>TRANS_REPETITION:long_name = "The repetition rate of transmission from the float";<br>TRANS_REPETITION:units = "second";<br>TRANS_REPETITION:_FillValue = 99999.f; | Repetition rate of the transmission system.<br>Unit : second<br>Example : 40 for a repetition of messages every 40 seconds. |
| POSITIONING_SYSTEM | char POSITIONING_SYSTEM(STRING8);<br>POSITIONING_SYSTEM:long_name = "Positioning system";<br>POSITIONING_SYSTEM:_FillValue = " "; | Position system from reference table 9.<br>ARGOS or GPS are 2 positioning systems.<br>Example : ARGOS |
| CLOCK_DRIFT | float CLOCK_DRIFT;<br>CLOCK_DRIFT:long_name = "The rate of drift of the float clock";<br>CLOCK_DRIFT:units = "decisecond/day";<br>CLOCK_DRIFT:_FillValue = "99999.f"; | Rate of drift of the float internal clock.<br>Unit : decisecond/day<br>Example : 1.57 |
| PLATFORM_MODEL | char PLATFORM_MODEL (STRING16);<br>PLATFORM_MODEL:long_name = "Model of the float ";<br>PLATFORM_MODEL:_FillValue = " "; | Model of the float.<br>Example :<br>APEX-SBE |
| PLATFORM_MAKER | char PLATFORM_MAKER (STRING256);<br>PLATFORM_MAKER:long_name = "The name of the manufacturer ";<br>PLATFORM_MAKER:_FillValue = " "; | Name of the manufacturer.<br>Example : Webb research |
| INST_REFERENCE | char INST_REFERENCE(STRING64);<br>INST_REFERENCE:long_name = "Instrument type";<br>INST_REFERENCE:conventions = "Brand, type, serial number";<br>INST_REFERENCE:_FillValue = " "; | References of the instrument : brand, type, serial number<br>Example : APEX-SBE 259 |
| WMO_INST_TYPE | char WMO_INST_TYPE(STRING4);<br>WMO_INST_TYPE:long_name = "Coded instrument type";<br>WMO_INST_TYPE:conventions = "Argo reference table 8";<br>WMO_INST_TYPE:_FillValue = " "; | Instrument type from WMO code table 1770.<br>A subset of WMO table 1770 is documented in the reference table 8.<br>Example :<br>846 : Webb Research float, Seabird sensor |

| | | |
|---|---|---|
| DIRECTION | char DIRECTION;<br>DIRECTION:long_name = "Direction of the profiles";<br>DIRECTION:conventions = "A: ascending profiles, B: descending and ascending profiles";<br>DIRECTION:_FillValue = " "; | Direction of the profiles of the float.<br>A : ascending profiles only<br>B : descending and ascending profiles |
| PROJECT_NAME | char PROJECT_NAME(STRING64);<br>PROJECT_NAME:long_name = "The program under which the float was deployed";<br>PROJECT_NAME:_FillValue = " "; | Name of the project which operates the profiling float that performed the profile.<br>Example : GYROSCOPE (EU project for Argo program) |
| DATA_CENTRE | char DATA_CENTRE(STRING2);<br>DATA_CENTRE:long_name = "Data centre in charge of float real-time processing";<br>DATA_CENTRE:conventions = "Argo reference table 4";<br>DATA_CENTRE:_FillValue = " "; | Code of the data centre in charge of the float data management.<br>The data centre codes are described in the reference table 4.<br>Example : ME for MEDS |
| PI_NAME | char PI_NAME (STRING64);<br>PI_NAME:comment = "Name of the principal investigator";<br>PI_NAME:_FillValue = " "; | Name of the principal investigator in charge of the profiling float.<br>Example : Yves Desaubies |
| ANOMALY | char ANOMALY(STRING256);<br>ANOMALY:long_name = "Describe any anomalies or problems the float may have had.";<br>ANOMALY:_FillValue = " "; | This field describes any anomaly or problem the float may have had.<br>Example : “the immersion drift is not stable.” |

## 2.4.4. Float deployment and mission information

| Name | Definition | Comment |
|---|---|---|
| LAUNCH_DATE | char LAUNCH_DATE(DATE_TIME);<br>LAUNCH_DATE:long_name = "Date (UTC) of the deployment";<br>LAUNCH_DATE:conventions = "YYYYMMDDHHMISS";<br>LAUNCH_DATE:_FillValue = " "; | Date and time (UTC) of launch of the float.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 03:05:00 |
| LAUNCH_LATITUDE | double LAUNCH_LATITUDE;<br>LAUNCH_LATITUDE:long_name = "Latitude of the float when deployed";<br>LAUNCH_LATITUDE:units = "degrees_north";<br>LAUNCH_LATITUDE:_FillValue = 99999.;<br>LAUNCH_LATITUDE:valid_min = -90.;<br>LAUNCH_LATITUDE:valid_max = 90.; | Latitude of the launch.<br>Unit : degree north<br>Example : 44.4991 : 44° 29' 56.76'' N |
| LAUNCH_LONGITUDE | double LAUNCH_LONGITUDE;<br>LAUNCH_LONGITUDE:long_name = "Longitude of the float when deployed";<br>LAUNCH_LONGITUDE:units = "degrees_east";<br>LAUNCH_LONGITUDE:_FillValue = 99999.;<br>LAUNCH_LONGITUDE:valid_min = -180.;<br>LAUNCH_LONGITUDE:valid_max = 180.; | Longitude of the launch.<br>Unit : degree east<br>Example : 16.7222 : 16° 43' 19.92'' E |
| LAUNCH_QC | char LAUNCH_QC;<br>LAUNCH_QC:long_name = "Quality on launch date, time and location";<br>LAUNCH_QC:conventions = "Argo reference table 2";<br>LAUNCH_QC:_FillValue = " "; | Quality flag on launch date, time and location.<br>The flag scale is described in the reference table 2.<br>Example :<br>1 : launch location seems correct. |
| START_DATE | char START_DATE(DATE_TIME);<br>START_DATE:long_name = "Date (UTC) of the first descent of the float.";<br>START_DATE:conventions = "YYYYMMDDHHMISS";<br>START_DATE:_FillValue = " "; | Date and time (UTC) of the first descent of the float.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 06 :05 :00 |
| START_DATE_QC | char START_DATE_QC;<br>START_DATE_QC:long_name = "Quality on start date";<br>START_DATE_QC:conventions = "Argo reference table 2";<br>START_DATE_QC:_FillValue = " "; | Quality flag on start date.<br>The flag scale is described in the reference table 2.<br>Example :<br>1 : start date seems correct. |
| DEPLOY_PLATFORM | char DEPLOY_PLATFORM(STRING32);<br>DEPLOY_PLATFORM:long_name = "Identifier of the deployment platform";<br>DEPLOY_PLATFORM:_FillValue = " "; | Identifier of the deployment platform.<br>Example : L'ATALANTE |
| DEPLOY_MISSION | char DEPLOY_MISSION(STRING32);<br>DEPLOY_MISSION:long_name = "Identifier of the mission used to deploy the float";<br>DEPLOY_MISSION:_FillValue = " "; | Identifier of the mission used to deploy the platform.<br>Example : POMME2 |
| DEPLOY_AVAILABLE_PROFILE_ID | char DEPLOY_AVAILABLE_PROFILE_ID(STRING256);<br>DEPLOY_AVALAIBLE_PROFILE_ID:long_name = "Identifier of stations used to verify the first profile";<br>DEPLOY_AVAILABLE_PROFILE_ID:_FillValue = " "; | Identifier of CTD or XBT stations used to verify the first profile.<br>Example : 58776, 58777 |
| END_MISSION_DATE | char END_MISSION_DATE (DATE_TIME);<br>END_MISSION_DATE:long_name = "Date (UTC) of the end of mission of the float";<br>END_MISSION_DATE:conventions = "YYYYMMDDHHMISS";<br>END_MISSION_DATE:_FillValue = " "; | Date (UTC) of the end of mission of the float.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 03:05:00 |
| END_MISSION_STATUS | char END_MISSION_STATUS;<br>END_MISSION_STATUS:long_name = "Status of the end of mission of the float";<br>END_MISSION_STATUS:conventions = "T:No more transmission received,<br>R:Retrieved";<br>END_MISSION_STATUS:_FillValue = " "; | Status of the end of mission of the float. |

### 2.4.5. Float sensor information

This section contains information about the sensors of the profiler.

| Name | Definition | Comment |
|---|---|---|
| SENSOR | char SENSOR(N_PARAM,STRING16);<br>SENSOR:long_name = "List of sensors on the float ";<br>SENSOR:conventions = "Argo reference table 3";<br>SENSOR:_FillValue = " "; | Parameters measured by sensors of the float.<br>The parameter names are listed in reference table 3.<br>Examples : TEMP, PSAL, CNDC<br>TEMP : temperature in celsius<br>PSAL : practical salinity in psu<br>CNDC : conductvity in mhos/m |
| SENSOR_MAKER | char SENSOR_MAKER(N_PARAM,STRING256);<br>SENSOR_MAKER:long_name = "The name of the manufacturer ";<br>SENSOR_MAKER:_FillValue = " "; | Name of the manufacturer of the sensor.<br>Example : SEABIRD |
| SENSOR_MODEL | char SENSOR_MODEL (N_PARAM,STRING256);<br>SENSOR_MODEL:long_name = "Type of sensor";<br>SENSOR_MODEL:_FillValue = " "; | Model of sensor.<br>Example : SBE41 |
| SENSOR_SERIAL_NO | char SENSOR_SERIAL_NO(N_PARAM,STRING16);<br>SENSOR_SERIAL_NO:long_name = "The serial number of the sensor";<br>SENSOR_SERIAL_NO:_FillValue = " "; | Serial number of the sensor.<br>Example : 2646 036 073 |
| SENSOR_UNITS | char SENSOR_UNITS(N_PARAM, STRING16);<br>SENSOR_UNITS:long_name = "The units of accuracy and resolution of the sensor";<br>SENSOR_UNITS:_FillValue = " "; | Units of accuracy of the sensor.<br>Example : psu |
| SENSOR_ACCURACY | float SENSOR_ACCURACY(N_PARAM);<br>SENSOR_ACCURACY:long_name = "The accuracy of the sensor";<br>SENSOR_ACCURACY:_FillValue = 99999.f; | Accuracy of the sensor.<br>Example : 0.005 |
| SENSOR_RESOLUTION | float SENSOR_RESOLUTION(N_PARAM);<br>SENSOR_RESOLUTION:long_name = "The resolution of the sensor";<br>SENSOR_RESOLUTION:_FillValue = 99999.f; | Resolution of the sensor.<br>Example : 0.001 |

### 2.4.6. Float calibration information

This section contains information about the calibration of the profiler. The calibration described in this section is an instrumental calibration. The delayed mode calibration, based on a data analysis is described in the profile format.

| Name | Definition | Comment |
|---|---|---|
| PARAMETER | char PARAMETER(N_PARAM,STRING16);<br>PARAMETER:long_name = "List of parameters with calibration information";<br>PARAMETER:conventions = "Argo reference table 3";<br>PARAMETER:_FillValue = " "; | Parameters measured on this float.<br>The parameter names are listed inreference table 3.<br>Examples : TEMP, PSAL, CNDC<br>TEMP : temperature in celsius<br>PSAL : practical salinity in psu<br>CNDC : conductvity in mhos/m |
| PREDEPLOYMENT_CALIB_EQUATION | char PREDEPLOYMENT_CALIB_EQUATION(N_PARAM,STRING256);<br>PREDEPLOYMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter";<br>PREDEPLOYMENT_CALIB_EQUATION:_FillValue = " "; | Calibration equation for this parameter.<br>Example :<br>Tc = a1 * T + a0 |
| PREDEPLOYMENT_CALIB_COEFFICIENT | char PREDEPLOYMENT_CALIB_COEFFICIENT(N_PARAM,STRING256);<br>PREDEPLOYMENT_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation"; | Calibration coefficients for this equation.<br>Example :<br>a1=0.99997 , a0=0.0021 |

| | | |
|---|---|---|
| | PREDEPLOYMENT_CALIB_COEFFICIENT:_FillValue = " "; | |
| PREDEPLOYMENT_CALIB_COMMENT | char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM,STRING256);<br>PREDEPLOYMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration";<br>PREDEPLOYMENT_CALIB_COMMENT:_FillValue = " "; | Comments applying to this parameter calibration.<br>Example :<br>The sensor is not stable |

## 2.4.7. Float cycle information

This section contains information on the cycle characteristics of the float. The values included in this section are programmed or estimated. They are not measured.

Each value has a N_CYCLES dimension. Each N_CYCLE describes a cycle configuration.

| Name | Definition | Comment |
|---|---|---|
| REPETITION_RATE | int REPETITION_RATE(N_CYCLES);<br>REPETITION_RATE:long_name = "The number of times this cycle repeats";<br>REPETITION_RATE:units = "number";<br>REPETITION_RATE:_FillValue = 99999; | Number of times this cycle repeats.<br>Usually, REPETITION_RATE and N_CYCLE are set to 1 : all the cycles are programmed to be the same.<br>However, some floats may perform cycles with different programming.<br>Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5th cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2.<br>The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1. |
| CYCLE_TIME | float CYCLE_TIME(N_CYCLES);<br>CYCLE_TIME:long_name = "The total time of a cycle : descent + parking + ascent + surface";<br>CYCLE_TIME:units = "decimal hour";<br>CYCLE_TIME:_FillValue = 99999.f; | Total time of a cycle.<br>This time includes the descending time, the parking time, the ascending time and the surface time.<br>Unit : decimal hour<br>Example : 240 hours for a ten day cycle. |
| PARKING_TIME | float PARKING_TIME(N_CYCLES);<br>PARKING_TIME:long_name = "The time spent at the parking pressure";<br>PARKING_TIME:units = "decimal hour";<br>PARKING_TIME:_FillValue = 99999.f; | Time spent at the parking pressure.<br>This time does not include the descending and ascending times.<br>Unit : decimal day<br>Example : 222 for 9 days and 6 hours at parking pressure. |
| DESCENDING_PROFILING_TIME | float DESCENDING_PROFILING_TIME(N_CYCLES);<br>DESCENDING_PROFILING_TIME:long_name = "The time spent sampling the descending profile";<br>DESCENDING_PROFILING_TIME:units = "decimal hour";<br>DESCENDING_PROFILING_TIME:_FillValue = 99999.f; | Time spent in descent.<br>Unit : decimal hour<br>Example : 8.5 for 8 hours 30 minutes of descending |
| ASCENDING_PROFILING_TIME | float ASCENDING_PROFILING_TIME(N_CYCLES);<br>ASCENDING_PROFILING_TIME:long_name = "The time spent sampling the ascending profile";<br>ASCENDING_PROFILING_TIME:units = "decimal hour";<br>ASCENDING_PROFILING_TIME:_FillValue = 99999.f; | Time spent in ascent.<br>Unit : decimal hour<br>Example : 7.5 for 7 hours 30 minutes of descending |
| SURFACE_TIME | float SURFACE_TIME(N_CYCLES);<br>SURFACE_TIME:long_name = "The time spent at the surface.";<br>SURFACE_TIME:units = "decimal hour";<br>SURFACE_TIME:_FillValue = 99999.f; | Time spent on the surface (surface drift).<br>Unit : decimal hour<br>Example : 10 for a 10 hours surface drift. |
| PARKING_PRESSURE | float PARKING_PRESSURE(N_CYCLES);<br>PARKING_PRESSURE:long_name = "The pressure of subsurface drifts";<br>PARKING_PRESSURE:units = "decibar";<br>PARKING_PRESSURE:_FillValue = 99999.f; | Pressure of the subsurface drift.<br>Unit : decibar<br>Example : 1500.0 for a subsurface drift at 1500.0 decibars. |
| DEEPEST_PRESSURE | float DEEPEST_PRESSURE(N_CYCLES);<br>DEEPEST_PRESSURE:long_name = "The deepest pressure sampled in the ascending profile";<br>DEEPEST_PRESSURE:units = "decibar";<br>DEEPEST_PRESSURE:_FillValue = 99999.f; | Deepest pressure sampled in the ascending profile.<br>Unit : decibar<br>Example : 2000.0 for an ascending profile starting at 2000.0 decibar. |
| DEEPEST_PRESSURE_DESCENDING | float DEEPEST_PRESSURE_DESCENDING(N_CYCLES);<br>DEEPEST_PRESSURE_DESCENDING:long_name = "The deepest pressure sampled in the descending profile";<br>DEEPEST_PRESSURE_DESCENDING:units = "decibar";<br>DEEPEST_PRESSURE_DESCENDING:_FillValue = 99999.f; | Deepest pressure sampled in the descending profile.<br>Unit : decibar<br>Example : 500.0 for a descending profile ending at 500.0 decibar. |

### 2.4.8. Highly desirable meta-data parameters

A highly desirable meta-data parameter should be correctly filled according to the following table.

| Highly desirable meta-data | mandatory format | example |
|---|---|---|
| DATA_TYPE | "Argo meta-data"; | DATA_TYPE = "Argo meta-data"; |
| FORMAT_VERSION | "2.2 "; | FORMAT_VERSION = "2.2 "; |
| HANDBOOK_VERSION | "1.2 "; | HANDBOOK_VERSION = "1.2 "; |
| DATE_CREATION | YYYYMMDDHHMISS | DATE_CREATION = "20040210124422"; |
| DATE_UPDATE | YYYYMMDDHHMISS | DATE_UPDATE = "20040210124422"; |
| PLATFORM_NUMBER | XXXXX or XXXXXXX | PLATFORM_NUMBER = "5900077 "; |
| PTT | not empty | PTT = "23978 "; |
| TRANS_SYSTEM | see reference table 10 | TRANS_SYSTEM = "ARGOS "; |
| TRANS_SYSTEM_ID | not empty | TRANS_SYSTEM_ID = "14281"; |
| POSITIONING_SYSTEM | see reference table 9 | POSITIONING_SYSTEM = "ARGOS"; |
| PLATFORM_MODEL | not empty | PLATFORM_MODEL = "SOLO"; |
| DIRECTION | "A" or "D" | DIRECTION = "A"; |
| DATA_CENTRE | see reference table 4 | DATA_CENTRE = "AO ; |
| LAUNCH_DATE | YYYYMMDDHHMISS | LAUNCH_DATE = "20010717000100"; |
| LAUNCH_LATITUDE | not empty, -90 <= real <= 90 | LAUNCH_LATITUDE = -7.91400003433228; |
| LAUNCH_LONGITUDE | not empty, -180 <= real <= 180 | LAUNCH_LONGITUDE = -179.828338623047; |
| LAUNCH_QC | see reference table 2 | LAUNCH_QC = "1"; |
| START_DATE | YYYYMMDDHHMISS | START_DATE = "20010702000000"; |
| START_DATE_QC | see reference table 2 | START_DATE_QC = "2"; |
| PARAMETER | see reference table 3 | PARAMETER ="PRES","TEMP","PSAL"; |
| CYCLE_TIME | not empty | CYCLE_TIME = 10; |
| DEEPEST_PRESSURE | not empty | DEEPEST_PRESSURE = 1092; |
| PARKING_PRESSURE | not empty | PARKING_PRESSURE = 1000; |

## 2.5. Technical information format

An Argo technical file contains technical information from an Argo float. This information is registered for each cycle performed by the float.

The number and the type of technical information is different from one float model to an other. To be flexible, for each cycle, the name of the parameters and their values are recorded. The name of the parameters recorded may therefore change from one model of float to another.

For file naming conventions, see §4.1 .

### 2.5.1. Dimensions and definitions

| Name | Definition | Comment |
|---|---|---|
| DATE_TIME | DATE_TIME = 14; | This dimension is the length of an ASCII date and time value.<br>Date and time values are always in universal time coordinates (UTC).<br>Date_time convention is : YYYYMMDDHHMISS<br>• YYYY : year<br>• MM : month<br>• DD : day<br>• HH : hour of the day<br>• MI : minutes<br>• SS : seconds<br>Examples :<br>20010105172834 : January 5th 2001 17:28:34<br>19971217000000 : December 17th 1997 00:00:00 |
| STRING256<br>STRING64,<br>STRING32<br>STRING16,<br>STRING8<br>STRING4<br>STRING2 | STRING256 = 256;<br>STRING64 = 64;<br>STRING32 = 32;<br>STRING16 = 16;<br>STRING8 = 8;<br>STRING4 = 4;<br>STRING2 = 2; | String dimensions from 2 to 256. |
| N_TECH_PARAM | N_TECH_PARAM = <int value> ; | Number of technical parameters.<br>Example :<br>N_TECH_PARAM=25<br>Twenty five different parameters are recorded for each cycle. |
| N_CYCLE | N_CYCLE = UNLIMITED; | Number of cycles performed by the float. |

## 2.5.2. General information on the technical data file

This section contains information about the technical data file itself.

| Name | Definition | Comment |
|---|---|---|
| PLATFORM_NUMBER | char PLATFORM_NUMBER(STRING8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII"; PLATFORM_NUMBER:_FillValue = " "; | WMO float identifier.<br>WMO is the World Meteorological Organization.<br>This platform number is unique.<br>Example : 6900045 |
| DATA_TYPE | char DATA_TYPE(STRING32); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file.<br>The list of acceptable data types is in the reference table 1.<br>Example : "Argo technical data" |
| FORMAT_VERSION | char FORMAT_VERSION(STRING4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " "; | File format version<br>Example : «2.2» |
| HANDBOOK_VERSION | char HANDBOOK_VERSION(STRING4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " "; | Version number of the data handbook.<br>This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook.<br>Example : «1.0» |
| DATA_CENTRE | char DATA_CENTRE(STRING2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " "; | Code of the data centre in charge of the float data management.<br>The data centre codes are described in the reference table 4.<br>Example : ME for MEDS |
| DATE_CREATION | char DATE_CREATION(DATE_TIME); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " "; | Date and time (UTC) of creation of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011229161700 : December 29th 2001 16 :17 :00 |
| DATA_UPDATE | char DATE_UPDATE(DATE_TIME); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file.<br>Format : YYYYMMDDHHMISS<br>Example :<br>20011230090500 : December 30th 2001 09 :05 :00 |

### 2.5.3. Technical data

This section contains a set of technical data for each profile.

There are N_TECH_PARAM (eg : 25) technical parameters recorded for each cycle.

For each cycle, for each technical parameter, the name of the parameter and the value of the parameter are recorded.

The parameter name and its value are recorded as strings of 32 characters.

The naming convention for TECHNICAL_PARAMETER_NAME is :

- Upper case letters only ;
- No space in the name (use underscore "_").

| Name | Definition | Comment |
|---|---|---|
| TECHNICAL_PARAMETER_NAME | char TECHNICAL_PARAMETER_NAME(N_CYCLE, N_TECH_PARAM, STRING32) TECHNICAL_PARAMETER_NAME:long_name="Name of technical parameters for this cycle"; TECHNICAL_PARAMETER_NAME:_FillValue = " "; | Name of the technical parameter.<br>Example :<br>"BATTERY_VOLTAGE"<br>See reference table 14 for standard technical parameter names. |
| TECHNICAL_PARAMETER_VALUE | char TECHNICAL_PARAMETER_VALUE(N_CYCLE, N_TECH_PARAM, STRING32) TECHNICAL_PARAMETER_VALUE:long_name="Value of technical parameters for this cycle"; TECHNICAL_PARAMETER_VALUE:_FillValue = " "; | Value of the technical parameter.<br>Example :<br>"11.5" |

Most parameter names are not standardized. However, certain usual parameter names are available in reference table 14.

## 2.6. GDAC FTP directory file format

### 2.6.1. Profile directory file format

The profile directory file describes all individual profile files of the GDAC ftp site. Its format is an autodescriptive Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

**Profile directory format definition**

# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all individual profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update

- file : path and file name on the ftp site. The file name contain the float number and the cycle number.
  Fill value : none, this field is mandatory
- date : date of the profile, YYYYMMDDHHMISS
  Fill value : " " (blank)
- latitude, longitude : location of the profile
  Fill value : 99999.
- ocean : code of the ocean of the profile as described in reference table 13
  Fill value : " " (blank)
- profiler_type : type of profiling float as described in reference table 8
  Fill value : " " (blank)
- institution : institution of the profiling float described in reference table 4
  Fill value : " " (blank)
- date_update : : date of last update of the file, YYYYMMDDHHMISS
  Fill value : " " (blank)

Each line describes a file of the gdac ftp site.

**Profile directory format example**

```
# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update
aoml/13857/profiles/R13857_001.nc,199707292003,0.267,-16.032,A,0845,AO,20030214155117
aoml/13857/profiles/R13857_002.nc,199708091921,0.072,-17.659,A,0845,AO,20030214155354
aoml/13857/profiles/R13857_003.nc,199708201845,0.543,-19.622,A,0845,AO,20030214155619
...
```

```
jma/29051/profiles/R29051_025.nc,200110250010,30.280,143.238,P,846,JA,20030212125117
jma/29051/profiles/R29051_026.nc,200111040004,30.057,143.206,P,846,JA,20030212125117
```

### 2.6.2. Trajectory directory format

The trajectory directory file describes all trajectory files of the GDAC ftp site. Its format is an autodescriptive Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

**Trajectory directory format definition**

```
# Title : Trajectory directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all trajectory files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 :  ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 :  ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, latitude_max, latitude_min, longitude_max, longitude_min, profiler_type, institution, date_update
```

- file : path and file name on the ftp site
  Fill value : none, this fiel is mandatory
- latitude_max, latitude_min, longitude_max, longitude_min : extreme locations of the float
  Fill values : 99999.
- profiler_type : type of profiling float as described in reference table 8
  Fill value : " " (blank)
- institution : institution of the profiling float described in reference table 4
  Fill value : " " (blank)
- date_update : date of last update of the file, YYYYMMDDHHMISS
  Fill value : " " (blank)

**Trajectory directory format example**

```
# Title : Trajectory directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all trajectory files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 :  ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 :  ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, latitude_max, latitude_min, longitude_max, longitude_min, profiler_type, institution, date_update
aoml/13857/13857_traj.nc,1.25,0.267,-16.032,-18.5,0845,AO,20030214155117
aoml/13857/13857_traj.nc,0.072,-17.659,A,0845,AO,20030214155354
aoml/13857/13857_traj.nc,0.543,-19.622,A,0845,AO,20030214155619
...
jma/29051/29051_traj.nc,32.280,30.280,143.238,140.238,846,JA,20030212125117
jma/29051/29051_traj.nc,32.352,30.057,143.206,140.115,846,JA,20030212125117
```

### 2.6.3. Meta-data directory format

The metadata directory file describes all metadata files of the GDAC ftp site. Its format is an autodescriptive Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

**Metadata directory format definition**

# Title : Metadata directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all metadata files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 :  ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 :  ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, profiler_type, institution, date_update

- file : path and file name on the ftp site
  Fill value : none, this field is mandatory
- profiler_type : type of profiling float as described in reference table 8
  Fill value : " " (blank)
- institution : institution of the profiling float described in reference table 4
  Fill value : " " (blank)
- date_update : date of last update of the file, YYYYMMDDHHMISS
  Fill value : " " (blank)

**Metadata directory example**

```
# Title : Metadata directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all metadata files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 :  ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 :  ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, profiler_type, institution, date_update
aoml/13857/13857_meta.nc,0845,AO,20030214155117
aoml/13857/13857_meta.nc,0845,AO,20030214155354
aoml/13857/13857_meta.nc,0845,AO,20030214155619
...
jma/29051/29051_meta.nc,846,JA,20030212125117
jma/29051/29051_meta.nc,846,JA,20030212125117
```

# 3. Reference tables

## 3.1. Reference table 1 : data type

The following table contains the list of acceptable contents for DATA_TYPE field.

| Name |
|---|
| Argo profile |
| Argo trajectory |
| Argo meta-data |
| Argo technical data |

## 3.2. Reference table 2 : Argo quality control flag scale

### 3.2.1. Reference table 2 : measurement flag scale

| n | Meaning | Real-time comment | Delayed-mode comment |
|---|---|---|---|
| 0 | No QC was performed | No QC was performed | No QC was performed |
| 1 | Good data | All Argo real-time QC tests passed. | The adjusted value is statistically consistent and a statistical error estimate is supplied. |
| 2 | Probably good data | Probably good data | Probably good data |
| 3 | Bad data that are potentially correctable | Test 15 or Test 16 or Test 17 failed and all other real-time QC tests passed. These data are not to be used without scientific correction. A flag '3' may be assigned by an operator during additional visual QC for bad data that may be corrected in delayed mode. | An adjustment has been applied, but the value may still be bad. |
| 4 | Bad data | Data have failed one or more of the real-time QC tests, excluding Test 16. A flag '4' may be assigned by an operator during additional visual QC for bad data that are not correctable. | Bad data. Not adjustable. |
| 5 | Value changed | Value changed | Value changed |
| 6 | Not used | Not used | Not used |
| 7 | Not used | Not used | Not used |
| 8 | Interpolated value | Interpolated value | Interpolated value |
| 9 | Missing value | Missing value | Missing value |

### 3.2.2. Reference table 2a : profile quality flag

*N* is defined as the percentage of levels with good data where:

- o QC flag values of 1, 2, 5, or 8 are GOOD data
- o QC flag values of 9 (missing) are NOT USED in the computation
- o All other QC flag values are BAD data

The computation should be taken from <PARAM_ADJUSTED_QC> if available and from <PARAM_QC> otherwise.

| n | Meaning |
|---|---|
| " " | No QC performed |
| A | ***N*** = 100%; All profile levels contain good data. |
| B | 75% <= ***N*** < 100% |
| C | 50% <= ***N*** < 75% |
| D | 25% <= ***N*** < 50% |
| E | 0% < ***N*** < 25% |
| F | ***N*** = 0%; No profile levels have good data. |

Example:

A TEMP profile has 60 levels (3 levels contain missing values).

- o 45 levels are flagged as 1
- o 5 levels are flagged as 2
- o 7 levels are flagged as 4
- o 3 levels are flagged as 9 (missing)

Percentage of good levels = ( (45 + 5) / 57) * 100 = 87.7%

PROFILE_TEMP_QC = "B";

## 3.3. Reference table 3 : parameter code table

The following table describes the parameter codes used for Argo data management.

| Code | Parameter long name | Comment | Unit | Valid min | Valid max | C_Format FORTRAN_ Format resolution | Fill value |
|---|---|---|---|---|---|---|---|
| CNDC | ELECTRICAL CONDUCTIVITY | In situ measurement | mhos/m | 0.f | 60.f | %10.4f F10.4 0.0001f | 99999.f |
| DOXY | DISSOLVED OXYGEN | In situ measurement | micromole/kg | 0.f | 650.f | %9.3f F9.3 0.001f | 99999.f |
| PRES | SEA PRESSURE | In situ measurement, sea surface = 0 | decibar | 0.f | 12000.f | %7.1f F7.1 0.1f | 99999.f |
| PSAL | PRACTICAL SALINITY | In situ measurement | psu | 0.f | 42.f | %9.3f F9.3 0.001f | 99999.f |
| TEMP | SEA TEMPERATURE IN | In situ measurement | degree_Celsius | -2.f | 40.f | %9.3f | 99999.f |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | SITU ITS-90 SCALE | | | | | F9.3<br>0.001f | |
| TEMP_DOXY | SEA TEMPERATURE FROM DOXY SENSOR (ITS-90 SCALE) | In situ measurement | degree_Celsius | -2.f | 40.f | %9.3f<br>F9.3<br>0.001f | 99999.f |

## 3.4. Reference table 4 : data centres and institutions codes

| Data centres and institutions | |
|---|---|
| AO | AOML, USA |
| BO | BODC, United Kingdom |
| CI | Institute of Ocean Sciences, Canada |
| CS | CSIRO, Australia |
| GE | BSH, Germany |
| GT | GTS : used for data coming from WMO GTS network |
| HZ | CSIO, China Second Institute of Oceanography |
| IF | Ifremer, France |
| IN | INCOIS, India |
| JA | JMA, Japan |
| JM | Jamstec, Japan |
| KM | KMA, Korea |
| ME | MEDS, Canada |
| NA | NAVO, USA |
| PM | PMEL, USA |
| RU | Russia |
| SI | SIO, Scripps, USA |
| SP | Spain |
| UW | University of Washington, USA |
| VL | Far Eastern Regional Hydrometeorological Research Institute of Vladivostock, Russia |
| WH | Woods Hole Oceanographic Institution, USA |

## 3.5. Reference table 5 : location classes (ARGOS)

| ARGOS location classes | |
|---|---|
| Value | Estimated accuracy in latitude and longitude |
| 0 | accuracy estimation over 1500m radius |
| 1 | accuracy estimation better than 1500m radius |
| 2 | accuracy estimation better than 500 m radius |
| 3 | accuracy estimation better than 250 m radius |
| G | GPS positioning |

### 3.6. Reference table 6 : data state indicators

| Level | Descriptor |
|---|---|
| 0 | Data are the raw output from instruments, without calibration, and not necessarily converted to engineering units. These data are rarely exchanged |
| 1 | Data have been converted to values independent of detailed instrument knowledge. Automated calibrations may have been done. Data may not have full geospatial and temporal referencing, but have sufficient information to uniquely reference the data to the point of measurement. |
| 2 | Data have complete geospatial and temporal references. Information may have been compressed (e.g. subsampled, averaged, etc.) but no assumptions of scales of variability or thermodynamic relationships have been used in the processing. |
| 3 | The data have been processed with assumptions about the scales of variability or hermodynamic relationships. The data are normally reduced to regular space, time intervals with enhanced signal to noise. |

| Class | Descriptor | Subclass |
|---|---|---|
| A | No scrutiny, value judgements or intercomparisons are performed on the data. The records are derived directly from the input with no filtering, or subsampling. | **-** Some reductions or subsampling has been performed, but the original record is available.<br><br>**+** Geospatial and temporal properties are checked. Geophysical values are validated. If not validated, this is clearly indicated. |
| B | Data have been scrutinized and evaluated against a defined and documented set of measures. The process is often automated (i.e. has no human intervention) and the measures are published and widely available. | **-** Measures are completely automated, or documentation is not widely available.<br><br>**+** The measures have been tested on independent data sets for completeness and robustness and are widely accepted. |
| C | Data have been scrutinized fully including intra-record and intra-dataset comparison and consistency checks. Scientists have been involved in the evaluation and brought latest knowledge to bear. The procedures are published, widely available and widely accepted. | **-** Procedures are not published or widely available. Procedures have not undergone full scrutiny and testing.<br><br>**+** Data are fully quality controlled, peer reviewed and are widely accepted as valid. Documentation is complete and widely available. |

#### Data state indicator recommended use

The following table describes the processing stage of data and the value to be assigned the data state indicator (DS Indicator). It is the concatenation of level and class described above.

| Processing Stage | DS Indicator |
|---|---|
| 1. Data pass through a communications system and arrive at a processing centre. The data resolution is the highest permitted by the technical constraints of the floats and communications system. | 0A (note 1) |
| 2. The national centre assembles all of the raw information into a complete profile located in space and time. | 1A (note 2) |
| 3. The national centre passes the data through automated QC procedures and prepares the data for distribution on the GTS, to global servers and to PIs. | 2B |
| 4. Real-time data are received at global data centres that apply QC including visual inspection of the data. These are then distributed to users in near real-time | 2B+ (note 3) |

| | |
|---|---|
| 5. Data are reviewed by PIs and returned to processing centres. The processing centres forward the data to the global Argo servers. | 2C |
| 6. Scientists accept data from various sources, combine them as they see fit with other data and generate a product. Results of the scientific analysis may be returned to regional centres or global servers. Incorporation of these results improves the quality of the data. | 2C+ |
| 7. Scientists working as part of GODAE generate fields of gridded products delivered in near real-time for distribution from the global servers. Generally, these products mostly will be based on data having passed through automated QC procedures. | 3B (note 4) |
| 8. Scientists working as part of GODAE generate fields of gridded products delivered with some time delay for distribution from the global servers. Generally, these products mostly will be based on data having passed through manual or more sophisticated QC procedures than employed on the real-time data. | 3C |

**Notes**

1. We need to have a pragmatic approach to what constitutes "original" or "raw" data. Despite the fact that an instrument may be capable of high sampling rates, what is reported from the instrument defines what is considered "raw". For example, Argo floats can certainly sample at finer scales than every 10 db, but because of communications, all we see for now is data at that (or worse) vertical resolution. Therefore the data "coming from the instrument" is "raw" output at 10db resolution.
2. The conversion of the raw data stream from the communications system into profiles of variables causes the data state indicator to switch from level 0 to 1.
3. Even though the data at global data centres use manual or semi-automated QC procedures, there is often not the intercomparisons to larger data collections and fields that would qualify the data state indicator to be set to class C. This is generally only provided by scientific scrutiny of the data.
4. The transition from class 2 to 3 occurs when assumptions of scales of variability are applied. During the course of normal data processing it is common to carry out some averaging and subsampling. This is usually done to exploit oversampling by the instrument, and to ensure good measurements are achieved. These are considered to be part of the geospatial and temporal referencing process.

## 3.7. Reference table 7 : history action codes

| Code | Meaning |
|---|---|
| CF | Change a quality flag |
| CR | Create record |
| CV | Change value |
| DC | Station was checked by duplicate checking software |
| ED | Edit a parameter value |
| IP | This history group operates on the complete input record |
| NG | No good trace |
| PE | Position error. Profile position has been erroneously encoded. Corrected if possible. |
| QC | Quality Control |
| QCF$ | Tests failed |
| QCP$ | Test performed |
| SV | Set a value |

| | |
|---|---|
| TE | Time error. Profile date/time has been erroneously encoded. Corrected if possible. |
| UP | Station passed through the update program |

## 3.8. Reference table 8 : instrument types

The instrument type codes comes from WMO table 1770. The WMO instrument types are available on the following web site :

[http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/J-COMM/CODES/wmotable_e.htm#ct1770](http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/J-COMM/CODES/wmotable_e.htm#ct1770)

| Code Figure | Instrument |
|---|---|
| 831 | P-Alace float |
| 840 | Provor, no conductivity |
| 841 | Provor, Seabird conductivity sensor |
| 842 | Provor, FSI conductivity sensor |
| 845 | Webb Research, no conductivity |
| 846 | Webb Research, Seabird sensor |
| 847 | Webb Research, FSI sensor |
| 850 | Solo, no conductivity |
| 851 | Solo, Seabird conductivity sensor |
| 852 | Solo, FSI conductivity sensor |
| 855 | Ninja, no conductivity sensor |
| 856 | Ninja, SBE conductivity sensor |
| 857 | Ninja, FSI conductivity sensor |
| 858 | Ninja, TSK conductivity sensor |

## 3.9. Reference table 9 : positioning system

| Code | Description |
|---|---|
| ARGOS | ARGOS positioning system |
| GPS | GPS positioning system |

## 3.10. Reference table 10 : transmission system

| Code | Description |
|---|---|
| ARGOS | Argos transmission system |
| IRIDIUM | Iridium transmission system |
| ORBCOMM | Orbcomm transmission system |

## 3.11. Reference table 11 : QC Test Ids

| ID | Test |
|---|---|
| 2 | Platform Identification test |
| 4 | Impossible Date test |
| 8 | Impossible Location test |
| 16 | Position on Land test |
| 32 | Impossible Speed test |
| 64 | Global Range test |
| 128 | Regional Global Parameter test |
| 256 | Pressure Increasing test |
| 512 | Spike test |
| 1024 | Top and Bottom Spike test |
| 2048 | Gradient test |
| 4096 | Digit Rollover test |
| 8192 | Stuck Value test |
| 16384 | Density Inversion test |
| 32768 | Grey List test |
| 65536 | Gross Salinity or Temperature Sensor Drift test |
| 131072 | Visual QC test |
| 261144 | Frozen profile test |
| 524288 | Deepest pressure test |

## 3.12. Reference table 12 : history steps codes

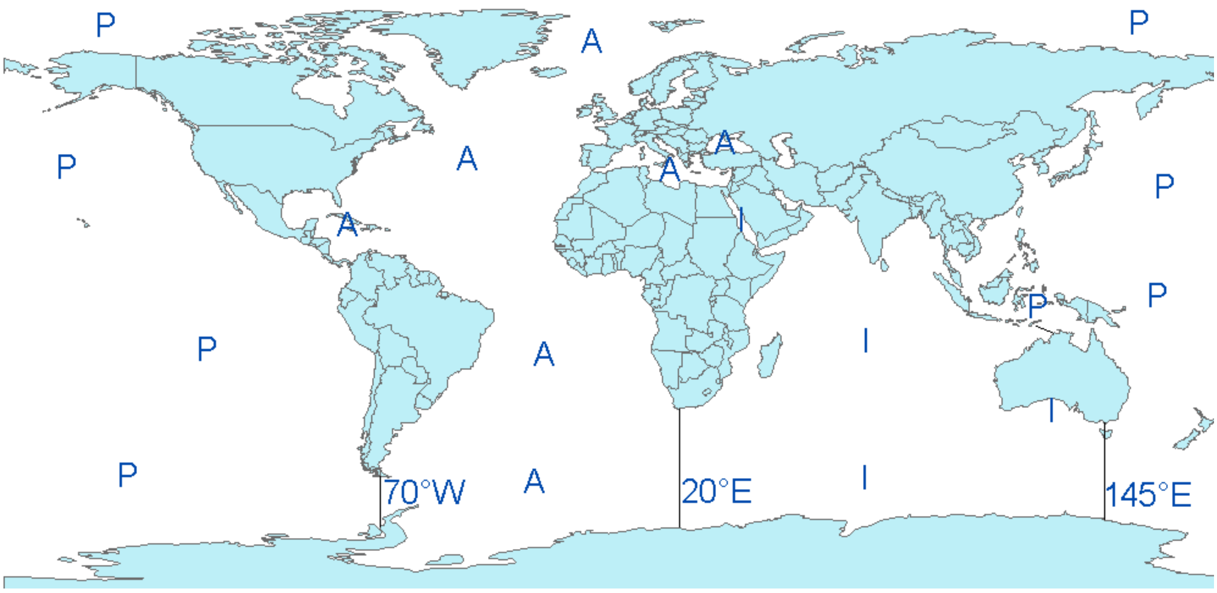
| Code | Meaning |
| --- | --- |
| ARFM | Convert raw data from telecommunications system to a processing format |
| ARGQ | Automatic QC of data reported in real-time has been performed |
| IGO3 | Checking for duplicates has been performed |
| ARSQ | Delayed mode QC has been performed |
| ARCA | Calibration has been performed |
| ARUP | Real-time data have been archived locally and sent to GDACs |
| ARDU | Delayed data have been archived locally and sent to GDACs |
| RFMT | Reformat software to convert hexadecimal format reported by the buoy to our standard format |

If individual centres wish to record other codes, they may add to this list as they feel is appropriate.

### 3.13. Reference table 13 : ocean codes

The ocean codes are used in the GDAC ftp directory files. The ocean code is not used in Argo NetCDF files.

| Code | Meaning |
|---|---|
| A | Atlantic ocean area |
| I | Indian ocean area |
| P | Pacific ocean area |




- The Pacific/Atlantic boundary is 70°W.
- The Pacific/Indian boundary is 145°E.
- The Atlantic/Indian boundary is 20°E.

### 3.14. Reference table 14 : technical parameter names

Most technical parameter codes are not standardized. However, certain usual parameter names are available.

| Parameter name | Meaning |
|---|---|
| SURFACE_PRESSURE | Sea surface pressure |
| BATTERY_VOLTAGE | Battery voltage in volts |

# 4. Data access

The whole Argo data set is available in real time and delayed mode from the global data centres (GDACs).

The internet addresses are :

- http://www.usgodae.org/argo/argo.html
- http://www.coriolis.eu.org/cdc

The FTP addresses are :

- ftp://usgodae1.fnmoc.navy.mil/pub/outgoing/argo
- ftp://ftp.ifremer.fr/ifremer/argo

The 2 GDACs offer the same data set that is mirrored in real time.

## 4.1. File naming convention on GDacs

The GADC ftp sites comply with the following naming conventions :

**Profile data**

- <FloatID>_prof.nc for a file containing all the profiles of a float.
  Example : 1900045_prof.nc
- <R/D><FloatID>_<xxx><D>.nc for an individual profile
  R for Real-Time and D for Delayed-Mode data
  xxx is the cycle number
  If the profile is a descending profile, D is added to the cycle number
  Examples : R1900045_003.nc, R1900045_003D.nc

**Trajectory Data**

- <FloatID>_traj.nc
  Example : 1900045_traj.nc

**Metadata**

- <FloatID>_meta.nc
  Example : 1900045_meta.nc

**Technical Data**

- <FloatID>_tech.nc
  Example : 1900045_tech.nc

## 4.2. Other data sources

All Argo data are available from Argo GDACs (Global data centres).

Some Argo data are also available from GTS (Global Telecommunication System), a network operated by WMO (World Meteorological Organization).

On GTS there are 2 formats for Argo profiles :

- TESAC : an Ascii format

- BUFR : a binary format under development.

The description of these format is available from the WMO web site :

- http://www.wmo.ch
- http://www.wmo.ch/web/www/DPS/NewCodesTables/WMO306vol-I-1PartA.pdf

# 5. Using the History section of the Argo netCDF Structure

Within the netCDF format are a number of fields that are used to track the progression of the data through the data system. This section records the processing stages, results of actions that may have altered the original values and information about QC tests performed and failed. The purpose of this document is to describe how to use this section of the format.

The creation of entries in the history section is the same for both profile and trajectory data. The next sections provide examples of what is expected. The information shown in the column labeled "Sample" is what would be written into the associated "Field" name in the netCDF format.

## 5.1. Recording information about the Delayed Mode QC process

The process of carrying out delayed mode QC may result in adjustments being made to observed variables. The table below shows how to record that the delayed mode QC has been done. Note that the fields HISTORY_SOFTWARE, HISTORY_SOFTWARE_RELEASE and HISTORY_REFERENCE are used together to document the name and version of software used to carry out the delayed QC, and the reference database used in the process. The contents of these three fields are defined locally by the person carrying out the QC.

Example: History entry to record that delayed mode QC has been carried out

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 |
| HISTORY_STEP | ARSQ | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | WJO | This is a locally defined name for the delayed mode QC process employed. |
| HISTORY_SOFTWARE_RELEASE | 1 | This is a locally defined indicator that identifies what version of the QC software is being used. |
| HISTORY_REFERENCE | WOD2001 | This is a locally defined name for the reference database used for the delayed mode QC process. |
| HISTORY_DATE | 20030805000000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | IP | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | FillValue | This field does not apply (1) |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | FillValue | This field does not apply |

**Note :**

(1) The present version of delayed mode QC only tests salinity and as such it is tempting to place "PSAL" in the _PARAMETER field. In future, delayed mode QC tests may include tests for temperature, pressure and perhaps other parameters. For this reason, simply addressing the software and version number will tell users what parameters have been tested.

## 5.2. Recording processing stages

Each entry to record the processing stages has a similar form. An example is provided to show how this is done. Note that reference table 12 contains the present list of processing stages and there should be at least one entry for each of these through which the data have passed. If data pass through one of these steps more than once, an entry for each passage should be written and the variable N_HISTORY updated appropriately.

Some institutions may wish to record more details of what they do. In this case, adding additional "local" entries to table 12 is permissible as long as the meaning is documented and is readily available. These individual additions can be recommended to the wider community for international adoption.

**Example**: History entry to record decoding of the data.

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 |
| HISTORY_STEP | ARFM | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | FillValue | This field does not apply |
| HISTORY_SOFTWARE_RELEASE | FillValue | This field does not apply |
| HISTORY_REFERENCE | FillValue | This field does not apply |
| HISTORY_DATE | 20030805000000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | IP | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | FillValue | This field does not apply |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | FillValue | This field does not apply |

## 5.3. Recording QC Tests Performed and Failed

The delayed mode QC process is recorded separately from the other QC tests that are performed because of the unique nature of the process and the requirement to record other information about the reference database used. When other tests are performed, such as the automated real-time QC, a group of tests are applied all at once. In this case, instead of recording that each individual test was performed and whether or not the test was failed, it is possible to document all of this in two history records.

The first documents what suite of tests was performed, and the second documents which tests in the suite were failed. A test is failed if the value is considered to be something other than good (i.e. the resulting QC flag is set to anything other than “1”). An example of each is provided. If data pass through QC more than once, an entry for each passage should be written and the variable N_HISTORY updated appropriately.

Example: QC tests performed and failed.

The example shown here records that the data have passed through real-time QC and that two tests failed. The encoding of tests performed is done by adding the ID numbers provided in reference table 11 for all tests performed, then translating this to a hexadecimal number and recording this result.

**Record 1**: Documenting the tests performed

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | FillValue | This field does not apply |
| HISTORY_SOFTWARE_RELEAS E | FillValue | This field does not apply |
| HISTORY_REFERENCE | FillValue | This field does not apply |
| HISTORY_DATE | 2003080500 0000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | QCP$ | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | FillValue | This field does not apply |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | 1BE | This is the result of all tests with IDs from 2 to 256 having been applied (see reference table 11) |

**Record 2**: Documenting the tests that were failed

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | FillValue | This field does not apply |
| HISTORY_SOFTWARE_RELEAS E | FillValue | This field does not apply |
| HISTORY_REFERENCE | FillValue | This field does not apply |
| HISTORY_DATE | 2003080500 0000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | QCF$ | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | FillValue | This field does not apply |
| HISTORY_START_PRES | FillValue | This field does not apply |

| | | |
|---|---|---|
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | A0 | This is the result when data fail tests with IDs of 32 and 128 (see reference table 11) |

## 5.4. Recording changes in values

The PIs have the final word on the content of the data files in the Argo data system. In comparing their data to others there may arise occasions when changes may be required in the data.

We will use the example of recomputation of where the float first surfaced as an example. This computation process can be carried out once all of the messages from a float have been received. Not all real-time processing centres make this computation, but it can be made later on and added to the delayed mode data. If this is the case, we would insert the new position of the profile into the latitude and longitude fields in the profile and we would record the previous values in two history entries. Recording these allows us to return to the original value if we have made an error in the newly computed position. The two history entries would look as follows.

**Example**: Changed latitude

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | FillValue | This field does not apply |
| HISTORY_SOFTWARE_RELEASE | FillValue | This field does not apply |
| HISTORY_REFERENCE | FillValue | This field does not apply |
| HISTORY_DATE | 20030805000000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | CV | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | LAT$ | A new entry for reference table 3 created by institution CI to indicate changes have been made in the latitude. |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | 23.456 | This is the value of the latitude before the change was made. |
| HISTORY_QCTEST | FillValue | This field does not apply |

**Notes** :

1. Be sure that the new value is recorded in the latitude and longitude of the profile section.
2. Be sure that the POSITION_QC flag is set to “5” to indicate to a user that the value now in the position has been changed from the original one that was there.
3. Be sure to record the previous value in history entries.

It is also sometimes desirable to record changes in quality flags that may arise from reprocessing data through some QC procedures. In this example, assume that whereas prior to the analysis, all temperature values from 75 to 105 dbars were considered correct, after the analysis, they are considered wrong. The history entry to record this would look as follows.

**Example**: Changed flags

| Field | Sample | Explanation |
|---|---|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12. |
| HISTORY_SOFTWARE | FillValue | This field does not apply |
| HISTORY_SOFTWARE_RELEASE | FillValue | This field does not apply |
| HISTORY_REFERENCE | FillValue | This field does not apply |
| HISTORY_DATE | 20030805000000 | The year, month, day, hour, minute, second that the process ran |
| HISTORY_ACTION | CF | Selected from the list in reference table 7 |
| HISTORY_PARAMETER | TEMP | Selected from the list in reference table 3 |
| HISTORY_START_PRES | 75 | Shallowest pressure of action. |
| HISTORY_STOP_PRES | 105 | Deepest pressure of action. |
| HISTORY_PREVIOUS_VALUE | 1 | This is the value of the quality flag on temperature readings before the change was made. |
| HISTORY_QCTEST | FillValue | This field does not apply |

**Notes** :

1. The new QC flag of "4" (to indicate wrong values) would appear in the <param>_QC field.